An EMAP Publication

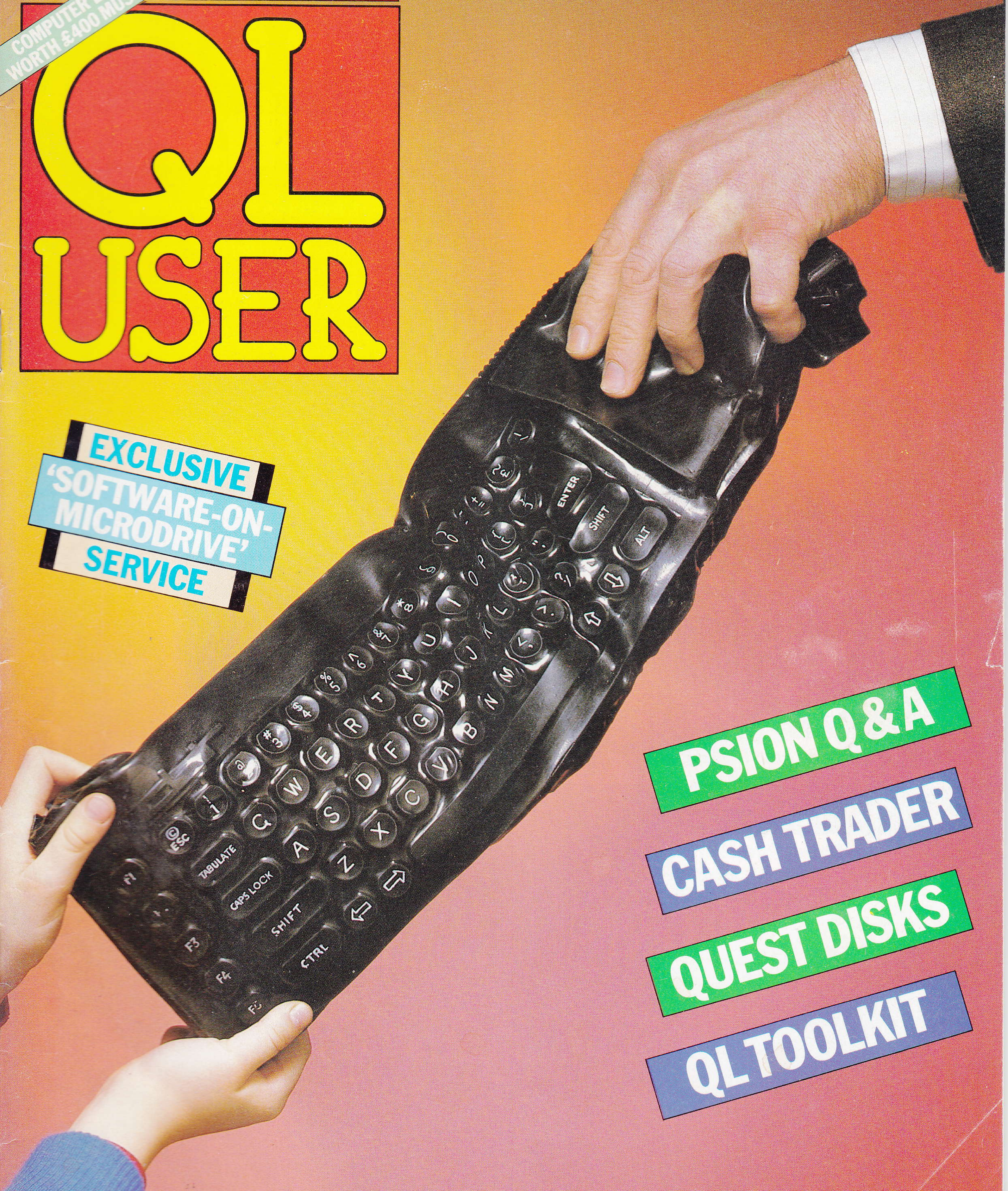
SINCL 'S SUPERMICRO

Every Month 95p MAY 1985

Germany DM5.50

COMPUTER LANGUAGES WORTH £400 MUST BE WON

QL USER



EXCLUSIVE 'SOFTWARE-ON-MICRODRIVE' SERVICE

PSION Q & A

CASH TRADER

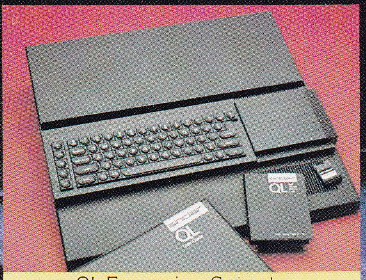
QUEST DISKS

QL TOOLKIT

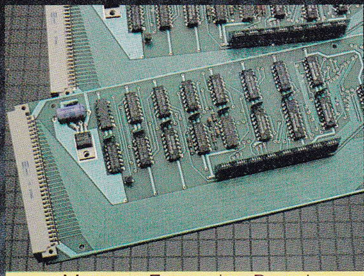
BUSINESS OR PLEASURE ?

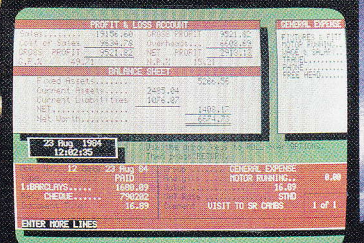
With Quest the QL really means business

Executive Series

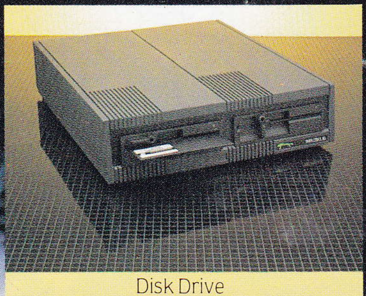


QL Expansion Console


Memory Expansion Board


```
PROFIT & LOSS ACCOUNT                              GENERAL EXPENSE
Sales........   19156.60  GROSS PROFIT   9521.82   FIXTURES & FITT
Cost of Sales    9634.78  Overheads....  6608.69   MOTOR RUNNING..
GROSS PROFIT     9521.82  NET   PROFIT   2913.13   WAGE & SALARY..
G.P.%      49.71          N.P.%    15.21           TRAVEL.........
                 BALANCE SHEET                     PHONE..........
   Fixed Assets.......                  5266.56    FREE HEAD......
   Current Assets.....      2485.04
   Current Liabilities      1076.87
   NET................                  1408.17
   Net Worth..........                  6674.73
 23 Aug  1984
   12:02:35       Use the arrow keys to ROLL the OPTIONS.
                  Then press RETURN.
          12      23 Aug 84               GENERAL EXPENSE
                       PAID               MOTOR RUNNING..        0.00
1:BARCLAYS.....     1680.09                         16.89
     CHEQUE......    790202                          STND
                      16.89         VISIT TO SR CAMBS           1 of 1
ENTER MORE LINES
```
QL Cashtrader*

CP/M 68K


Disk Drive

*Available from QL stockists or direct from
Sinclair Research Ltd.,
Camberley (0276) 68311

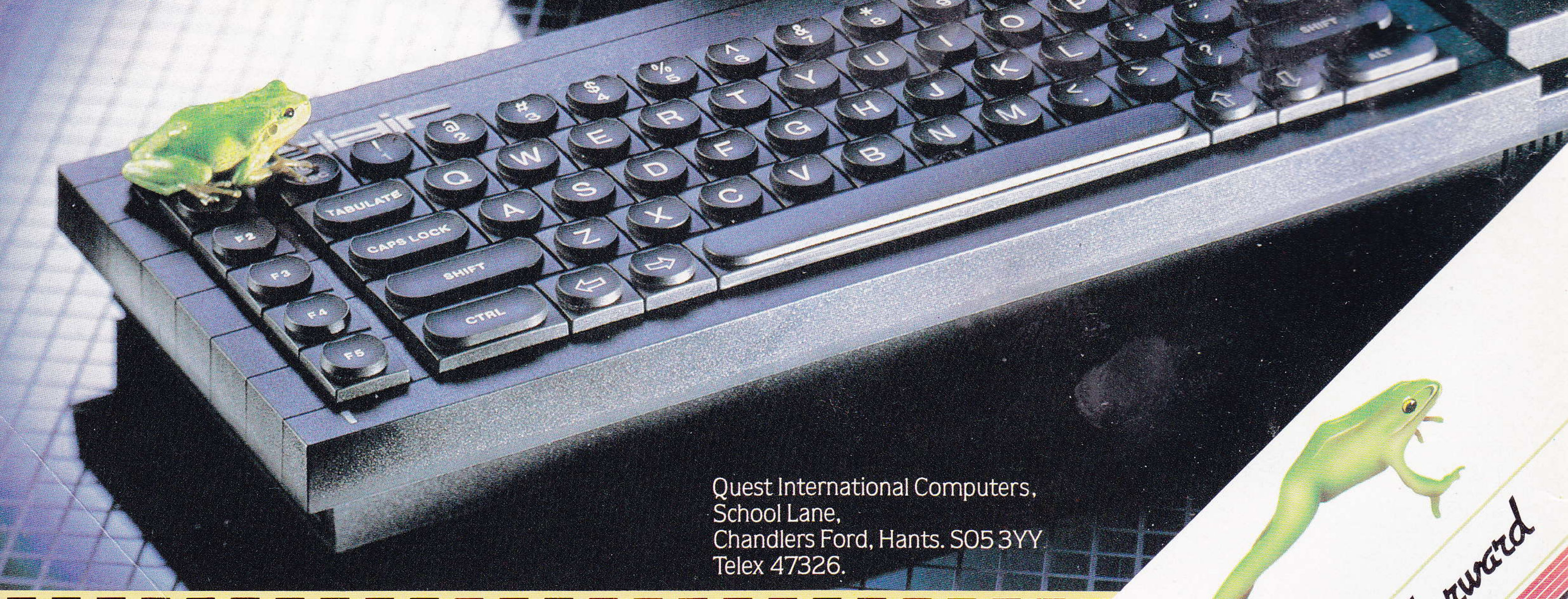
Quest's Executive series of quality peripherals and software for the Sinclair QL allows the business and serious home user to fully realise the full potential of the most exciting computer of the 80's.

Post the coupon below to obtain details of a wide range of Quest Executive products including:

- CP/M 68K
- Business Accounting Software
- Floppy and Winchester Disk Drives
- Memory Expansion Boards
- Monitors
- Printers
- Consumables

Telephone Sales Desk 04215 66488

CP/M is a registered trademark of Digital Research
QL is a registered trademark of Sinclair Research



Quest International Computers,
School Lane,
Chandlers Ford, Hants. SO5 3YY
Telex 47326.

Please send me details of the Quest Executive Series

Name ______________________ Tel. No. ______________________

Address ______________________ My QL Serial Number

______________________ is ______________________

Position ______________________ QLU5

Another leap forward

quest

QL USER

May 1985

Editor	Paul Coster BSc
Assistant Editor	Paolo Baccanello
Editorial Assistant	Shirley Eborn
Art Editor	Mike Spiller
Technical Consultant	Adam Denning
Associate Editor	Peter Rodwell
Advertising Manager	Phil Baker
Advert Production	Yvonne Moyser
Publisher	Terry Pratt

Contributors

Sid Smith, Leon Heller, Nicky Trevett, Adam Denning, Giles Todd, Roger Vernon, Richard Cross, Ian Williams, Jane Robins, Simon Craven, Irwin Joffe, Mike James, R Shergold, D Tose and Stephen Hollywood

Editorial 01-251 6222

QL User, Priory Court,
30-32 Farringdon Lane,
London EC1R 3AU.

Advertising 01-251 6222

Model by Mike Pope, photo by Rob Brimson

Competitions

Not an announcement of winners this time, but an appeal to entrants of the Talent Spot competition. Believe it or not, two readers have sent in *blank* drives as entries for this competition – one envelope postmarked 'Romford', the other too blurred to read. The 'Romford entry' came in a brown envelope with the drive labelled 'Competition'. The other arrived in a white envelope and the drive was unmarked and unformatted (!)

So, if you've entered this competition and not had your drive returned, it might be prudent to give us a ring – we'll be judging the competition on 22nd April.

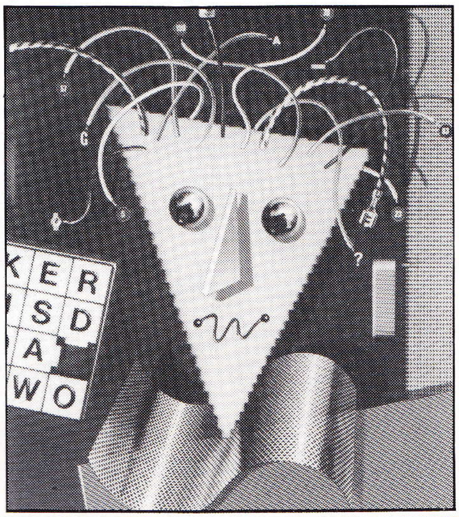
Information

There appears to have been some confusion surrounding April's Lunar Lander program. The original listing was intended for disk (hence the reference to 'flp'), but it's a simple matter to convert it to microdrive. Just change 'flp' to 'mdv' in line 330.

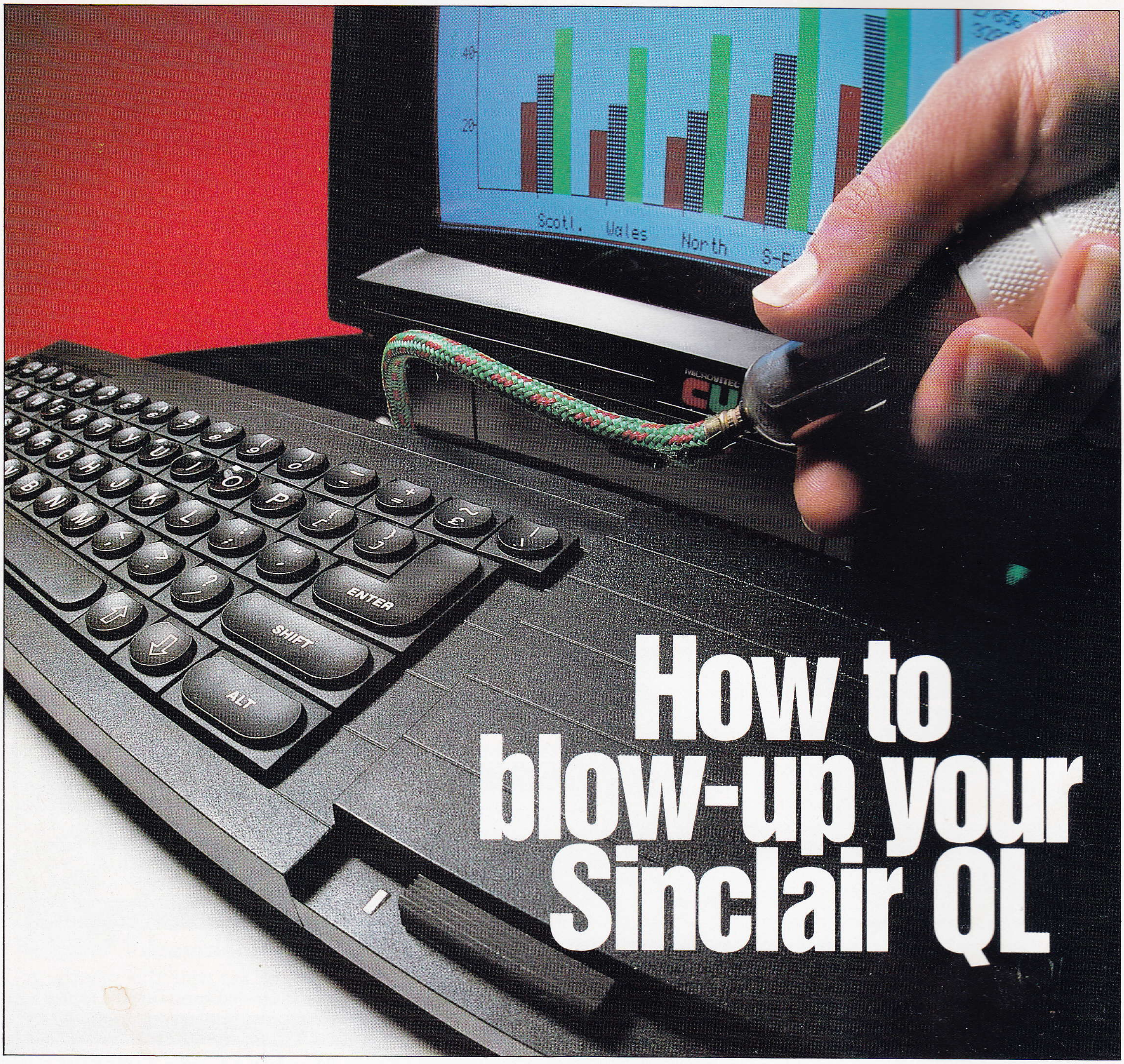
Medic Malaise

Several readers have voiced concern about the company Medic Datasystems Ltd with regard to non-delivery of disk systems advertised in QL User and at recent microcomputer shows. After conversations with the company in question we would like to make the following points: QL User has not yet *reviewed* any Medic products, though we *have* recently seen one of the 512K combined disk interface and expansion cards along with dual 720K drives. We have also been assured that production units will be sent out starting 12th April 1985. Readers who placed orders with Medic more than 28 days ago and have still not received the goods can voice their concern on 0256 460748 – and don't forget that anyone has a right to their money back if goods are not delivered within the period originally specified or are not as described within any associated advertising.

CONTENTS



Published and Distributed by EMAP Business and Computer Publications Ltd.
Typesetting by Contemporary Graphics Ltd., 20 Bowling Green Lane, London EC1.
Printing Riverside Press.



First, fuse together your Sinclair QL with **QL/APL**. Immediately you've expanded your QL's potential to the maximum.

The unique **QL/APL** programming language frees you from the constraints of loops and endlessly long programs. **QL/APL** lets the computer cope effortlessly with data types and numeric precision and all that technical stuff. Leaving you free to get your QL to work for you, not against you.

QL/APL is easy to work with, so it's simple to write your own applications.

QL/APL is modular, so you can build up systems step-by-step.

QL/APL is interpreted, so changes can be made and tested immediately.

QL/APL is incredibly powerful, so you never come up against annoying limitations. It's the key to taking advantage of the full 32-bit power of the QL's 68008 processor. It runs on the standard QL keyboard, and allows full access to QL colour graphics and windows. And it's built around MicroAPL's highly- respected IBM-compatible implementation of the APL language – in fact APL was originally developed by IBM for its mainframes.

A software pedigree you can trust implicitly.

MICRO
APL

Remember, for QL, read QL/APL. And blow the rest

To: MicroAPL Ltd, Unit 1F, Nine Elms Industrial Estate,
87 Kirtling Street, London SW8 5BP.

□ Please send me more details about the power of **QL/APL**.
□ Please send me _____copies of **QL/APL** at £99.95 plus £1.95 p&p
□ I enclose cheque/P.O. made payable to MicroAPL Ltd.
□ Please charge my Visa/Access card number __________
Card expiry date__________

Name: __________

Address: __________

____

____

Tel No __________

QL/5/85

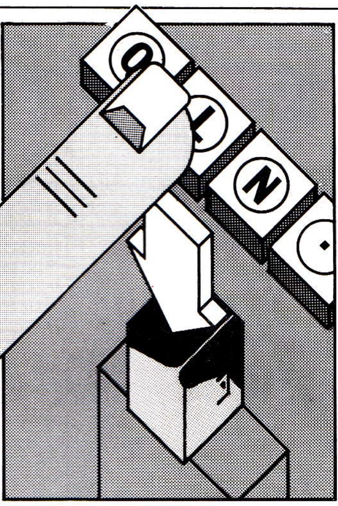
QL and Sinclair are trademarks of Sinclair Research Ltd.

(Prices include VAT. Please allow 28 days for delivery. For credit card telephone orders, phone 01-622 0395).

NEWS

Hard hitting news from QL User's bullish

investigative reporter, Sid Smith.



Whatever Will Be . . .

Sinclair's MD Nigel Searle has promised to ensure that the QL's QCOM telecommunication device will reach the market – despite the receivership of its manufacturer, OE Ltd.

"Whatever happens to OE as a company, those products will still be available to QL owners – whether Sinclair will buy stocks from OE, if they exist, and sell them, whether Sinclair will buy a license to produce those products, or whether Sinclair will act as an honest broker and ensure that some third party acquires a license to produce them.

"They are a valuable asset to the QL, and I'm certain that we can find a way to get them produced."

OE boss Martin Ansell said that the receivers had been called after his company had received "an unmeetable bill' from chief shareholders Warburtons the bankers. He hoped that OE would be sold as a going concern, rather than split into occasionally intangible assets like QCOM.

At the time of writing, the future of OE has yet to be decided, though at least one large concern (not Sinclair Research!) has told us of its interest.

Further proof of the importance which Sinclair attribute to QCOM is supplied, not merely by its prominence in their glossy advertising supplements, but also by what sounds like an underwriting of its production costs.

According to Searle, "We had a contract with OE which put Sinclair in a position where we were obliged to promote these products and in effect to guarantee certain minimum sales, and we'd be willing to make a similar contractual arrangement with some other company that took over those products."

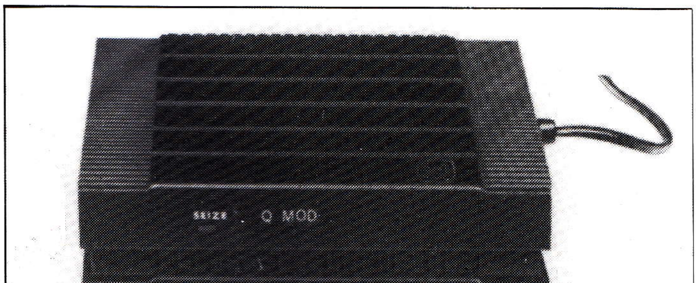
As reported last month, some QL users have been waiting since late November for their QCOM orders to be filled – OE having cashed their cheques on receipt. The position of these unfortunates now looks even gloomier.

Pascal Perfidy

Following copious two-way correspondence with Computer One, it seems that we may have been a little over zealous in our review of their Pascal.

The purist's approach – it either *is*, or it *isn't* standard Pascal – is a mite pedantic where certain useful extensions have been included (as with Computer One's).

Suffice it to say that our points were aimed at the language in general, not to any particular version, and that all comments of this nature are likely to be coloured by the reviewer's favourite language at the time.



The ill-fated OEL modem.

The much-publicised Atari 520 ST.

Attack On Atari

"Clive Sinclair doesn't think you'll produce the Atari ST on time," your reporter told Jack Tramiel.

Replied the bullfrog-like Atari boss, "Who?"

Once we'd got past this too-familiar Tramiel joke (he even professes ignorance of Commodore, the company he steered to fortune and then left), it turned out that Sinclair, and even QL, were names he *had* come accross before.

No, he wasn't worried about competition from the QL, which was "a terrific machine when it was launched, but out of date by the time it appeared".

He didn't want to criticise Sir Clive ("In case I ever run into him in a Cambridge pub"), but nobody had heard of the QL in the States, and "from what I understand it hasn't been tremendously successful here in England".

Confronted with this Tramiel-talk, Sir Clive was equally agressive. "I bet you a hundred quid to ten that the ST won't appear on time. Not a chance. When Tramiel announced the machine in Las Vegas he said he needed $180m to develop it – and he hasn't raised a penny."

"In any case," continued Clive, "the GEM software (Digital Research's Macintosh-style user interface, to be employed in the ST) won't be available till late this year, so the Atari machine can't appear before then."

Lots of people, we suggested, assumed that if the ST appeared, it would kill all possibility of selling the QL in the States.

"But why? The ST hasn't even appeared yet, it has no software, and it doesn't have drives built in. No, my bet is that we'll launch the QL in the States long before the ST appears and it'll sell in large numbers.

"I don't think the ST will appear at all. Jack's a great guy, but he's a little prone to announcing products before they're ready – perhaps like some of the rest of us – and this time he's exceeded himself. I don't think he's got the means to produce that product."

ROM Deal

The mystery of the JS ROM (see our last issue) seems to be solved; the new firmware, along with its extra SuperBasic commands, was all a mistake.

JS was only ever shipped to a handful of people, all of them previous owners returning faulty QLs for repairs.

In a commendable desire to speed up turn-round, the company – as usual – replaced rather than repaired the faulty machines. On this occasion, however, despatch clerks managed to ship QLs being used to test a development operating system, code-named JS.

Sinclair are very shy about the affair, but we understand that no new version of QDOS is expected for several months.

CUB AND QL– THE PERFECT PARTNERSHIP

```
command> View figures with format? 7
Format 7 Rep PIE
Current Name figures
                                        12K Memory
```

This Cub medium resolution colour monitor is TOTALLY compatible with the Sinclair QL.

It combines outstanding graphics capabilities with the facility for displaying full 85 column text whilst doing full justice to the colour potential of the QL. Its new cabinet with Tilt and Swivel Plinth harmonises perfectly with the QL's simple yet functional appearance.

Best of all, a price of only

£ 295 Inc VAT

(inc. Tilt and Swivel Plinth) and VAT keeps your bank balance in the black too.

SPECIFICATION

Model – CUB 1451/DQ3
14" QL monitor
RGB/TTL input
Tube Resolution (pixels) – 653 (H)
Pitch – 0.43 mm
Bandwidth – 18 MHz
Antiglare CRT

MICROVITEC
cub
COLOUR MONITORS

Microvitec plc., Futures Way, Bolling Road,
Bradford, West Yorkshire BD4 7TU.
Tel: (0274) 390011/726500. Telex: 517717

1984

AVAILABLE FROM HIGH STREET COMPUTER RETAILERS AND BRANCHES OF W. H. SMITH, BOOTS, JOHN LEWIS PARTNERSHIP, LASKEYS, CURRYS.

THE NAME 'CUB' IS A REGISTERED TRADE MARK OF MICROVITEC PLC.

One year on and a QL dilemma — strive for

BUSINESS OR PLEASURE?

BUSINESS OR PLEASURE?

PHOTO ROB BRIMSON — MODEL BY MIKE POPE

business status or dominate the home market. 'Times' columnist, Simon Craven, assesses developing trends.

Much has been written and said about the near hysteria that swept through the ranks of normally cycnical journalists at the original launch of the QL over a year ago. Articles predicting the downfall of IBM were hastily penned. Shares in ACT would have been sold if your average hack had been rich enough to own any in the first place. It even got to the stage where magazines were launched.

Shortly afterwards came the equally dramatic backlash, made all the worse by the degree to which the original coverage went over the top. The QL failed to appear for some considerable time and even when it did finally see the light of day the results were far removed from what we had seen at the grand launch. A certain degree of terminological inexactitude on Sinclair Research's part was suspected. Consequently the £400 chocolate bar lost a large degree of street credibility and was declared by the pundits (very few of whom, it should be noted, had actually used the thing) as not possessing 'the right stuff' to be a hard-nosed serious machine suitable for hard-nosed serious applications.

This identity crisis was not helped by Sinclair's rival manufacturers. For the next three months, anyone who launched any kind of computer was asked "What do you think of the QL". The home computer people replied that the QL was not in their market – it was clearly a business machine, look at the software they supply, and anyway we don't regard it as competition. Trained observers could usually detect a slight quickening of the pulse and glistening of the speaker's brow at this stage. The business people took the same attitude in reverse. "The QL is a very nice little machine", they said. "I'm thinking of getting one for the kids to play with. Meanwhile, would you care to examine our new machine, with real disk drives, available ex-stock from our dealer network?"

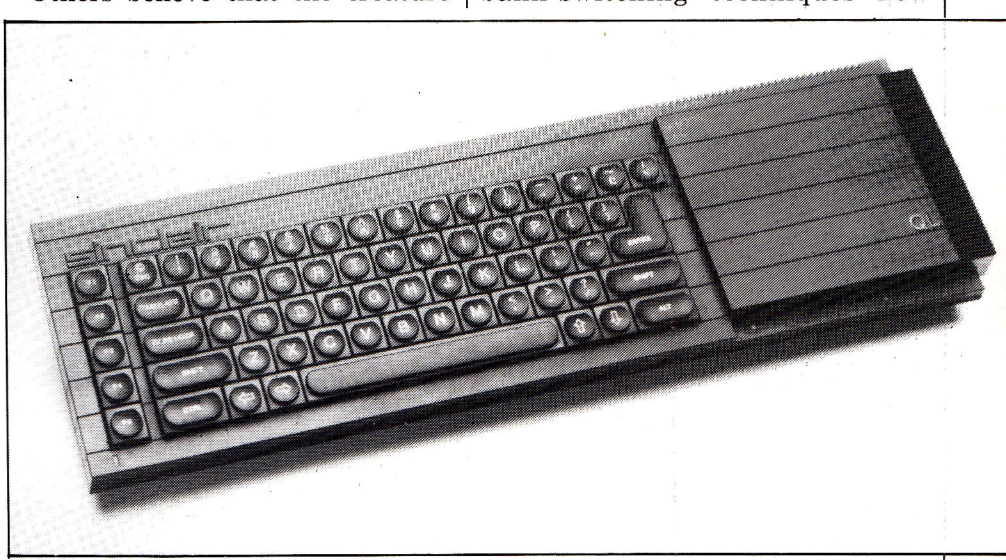
Over a year after customer deliveries started, the QL is still a difficult machine to position in the microcomputer market. In a similar kind of price range are machines like the BBC Micro and the Amstrad CPC464, whereas in terms of raw horsepower and potential for sophisticated applications software the alternatives have to be drawn from much further up-market.

Private Investigations

Finding equivalents to the QL's four Psion programs is easy enough – a slightly souped up version called Xchange is available for the IBM PC and ACT Apricot among other Intel 8086/8088 based MS-DOS machines. However, these micros are perceived as being in a different class to the QL – whether or not they should be is a question we will come to shortly.

Finding out where the QL is really intended to be in the market and who the target buyer is, is not easy. Noone will say anything definite, perhaps because they are frightened of restricting the machine's appeal by putting it in a pigeon-hole. If, however, you get a Sinclair Research spokesman, a big-time QL software developer or a typical press pundit in a quiet corner and threaten him with a very large stick, you might just get some comments about "home professional" users.

The home professional user is a semi-mythical creature like the Loch Ness Monster or Bigfoot – no-one is sure if he exists or not and since there are no specimens in captivity little if anything is known about his computer needs. Some independent observers claim that the home professional user is a creation of overimaginative marketing executives, the product of wishful thinking from home computer manufacturers whose market has inconveneintly stopped doubling once a fortnight.



Others believe that the creature does exist, but is an endangered species, trapped between the sparse pickings of the BBC Micro and the vast admission fee to the lush IBM PC meadow.

Are there large numbers of people who want to do serious things with computers in the comfort and privacy of their own homes? Probably. If you look at the Spectrum software charts, one of the most consistent sellers is a word processor, Tasword Two. Tasword does the best it can, but you can't get very far with a Spectrum as a word processor. Still, the demand is obviously there, even among users of the most games-orientated computers. The QL, of course, has a high quality word processor thrown in free, but among the business programs shown to the press by independent software vendors a few weeks ago were several more specialised business programs which first appeared on machines like the Amstrad and the BBC, and in some cases even the Spectrum!

The QL is well suited as a home professional machine, and to judge by the expansion systems, peripherals and software becoming available, it is this role which attracts most of the third party manufacturers with an eye on the QL bandwagon. With this in mind, it can be seen as a threat to an enormous range of different machines, and it is interesting to examine six of the best, to see how they stack up against the Cambridge Conundrum.

Six Of The Best

At the lower end of the comparison are two successful home micros which have grown up into useful systems from fairly humble roots. The BBC Micro and Amstrad CPC464 are both eight-bit machines, with the 6502 and Z80 processors respectively.

These processors have only sixteen-bit address busses which inevitably limit the amount of RAM which can be used. Although clever bank-switching techniques now allow the use of more than 64K, with screen RAM in a separate area and the ROM overlaid on to RAM only when needed, it is still impractical to try to use more than 64K for any one program.

Being Z80 based, and blessed with a reasonably user-friendly manufacturer, the Amstrad has recently sprouted the CP/M operating system for use with its compact 3" disk drive, and a reasonable amount of CP/M software has already been ported over. Don't think, however, that you will be able to run anything in the (admittedly huge) CP/M software world, though. Apart from the problem of buying the stuff on the right disk format, the Amstad under CP/M has a smaller TPA (transient program area) than many other machines and a number of programs run short of elbow room as a result.

The QL comes next in the lineup – naturally this is a known quantity for most readers, but in this context it is worth noting that the nice thing about sixteen-bit processors isn't the speed, it's the amount of memory which can be accessed directly. This is especially true when running business style programs, in which case the limiting factor on the time taken to finish a job is not the speed of the processor but the speed of input and output (including to and from mass storage devices).

The relative unimportance of processing speed enabled Sinclair to use a low-cost solution with the 68008 member of the 68000 family, along with eight-bit supporting architecture.

Slightly further up-market we have the Apple IIc, an interesting device which represents the nearest thing in computer terms to progress through evolution rather than revolution. The Apple II was one of the first three practical desktop micros built, it has been kept reasonably fit with regular injections of monkey glands. The IIc is a 6502-based machine – no need to poke fun just because of that – with 128K in two banks of 64K which are switched in such a way as to make them fairly transparent to the non-programming user. The best use is probably as a data storage area to keep down use of the rather arthritic disk drive.

Like the QL, the Apple IIc is semi-portable. It is highly compact, with the mass storage built into the keyboard unit, and the recommended monitor is small enough to carry around in the other hand (try that with your Vision QL!) The operating system is Apple's own PRODOS, which runs most existing Apple software written under Apple DOS 3.3.

A bit more potent in terms of showroom appeal is another Apple machine, the Macintosh. QL apart, the Mac is the only 68000-based machine available at less than the price of a fairly reasonable car, at least until such time as Atari and/or Commodore actually turn words into deeds and put their low-cost 68K systems on sale.

The QL and the Mac are similar in other ways. Apple and Sinclair share a feel for bold design and an ability to guess what the potential buyer wants. Neither sets much (if any) store by industry standards, preferring to go their own way if they can see how to speed up a process here or save a buck there. The Mac goes completely outside normal conventions with its graphics-based user interface, and runs no recognised operating system – it just is.

Rounding off, we have two computers built around the Intel 8086/8088 chip, both running MS-DOS, or if you prefer, CP/M-86. That's where the similarities end between the Sanyo MBC550 and the ACT Apricot F1. The Apricot is an idiosyncratic design job, with a built-in trendy 3½″ disk drive and a fancy infra-red keyboard, whereas the Sanyo is a thoroughly conventional machine looking like an IBM PC on the Scarsdale Diet.

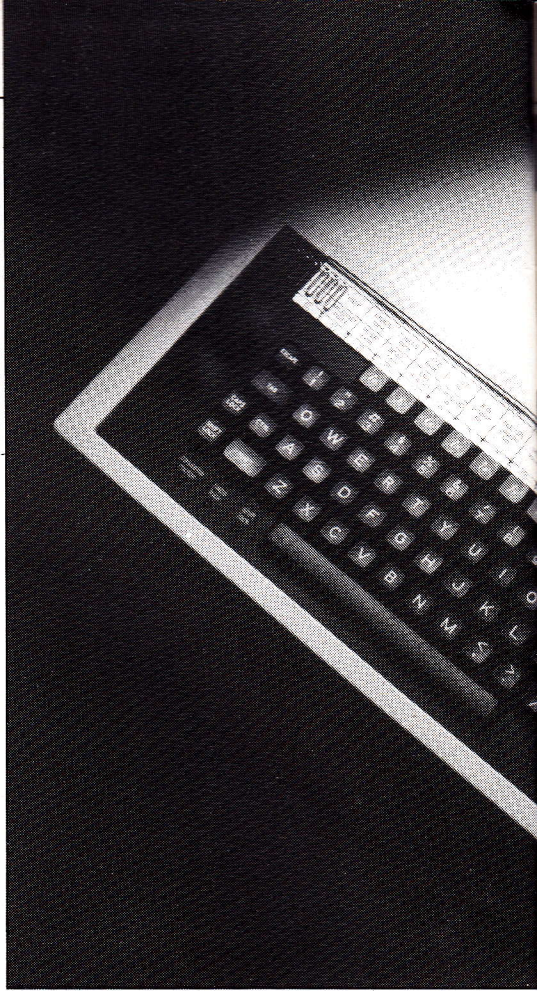
Ergonomics And Design

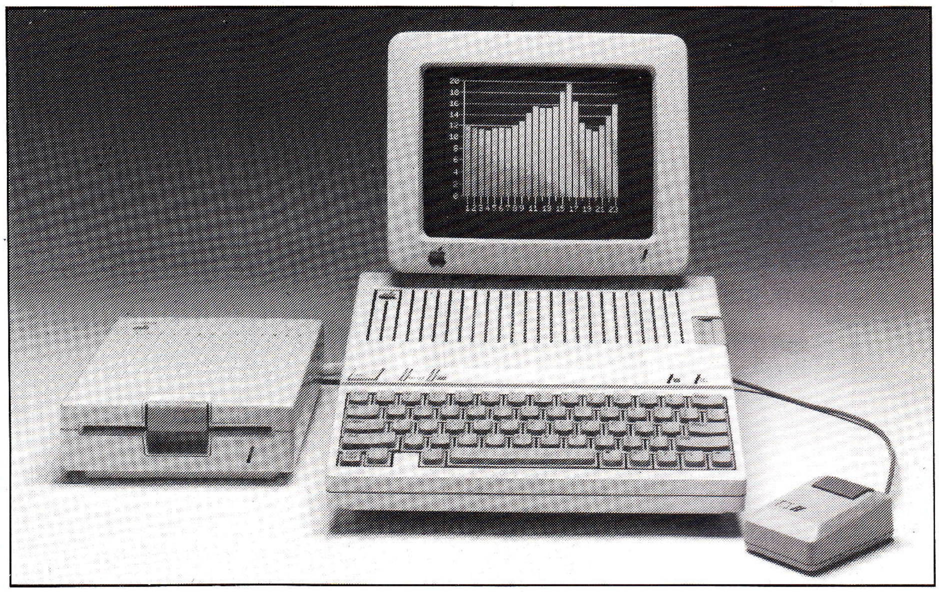
It is surprising how many computers would qualify for consideration as home/professional machines if they were improved in just two areas, mass storage and ergonomics. For any kind of serious use, the cassette recorder is out of the question, and if you are going to spend all day looking at a screen, that old standby the 14″ colour portable is going to have to give way to something which will display reasonable 80-column text. Similarly, there is little joy to be had by spending hours on end typing on a keyboard which feels as if the keys are mounted on suet.

The QL has recently been improved in all these areas. The latest keyboards are reasonable, though still subject to personal taste. There are plenty of high quality RGB monitors, of which the official version is probably the best, and the mass storage question has been answered by the CST floppy disk interface amongst others. The speed increase brought about by connecting a pair of disk drives to the QL is remarkable – Quill overlays load almost instantly, program compilation time is halved and that irritating feeling of 'waiting for the computer' disappears. You can hang any BBC Micro disk drive on to the CST unit, and perhaps the best choice would be a pair of the 3½″ Sony type.

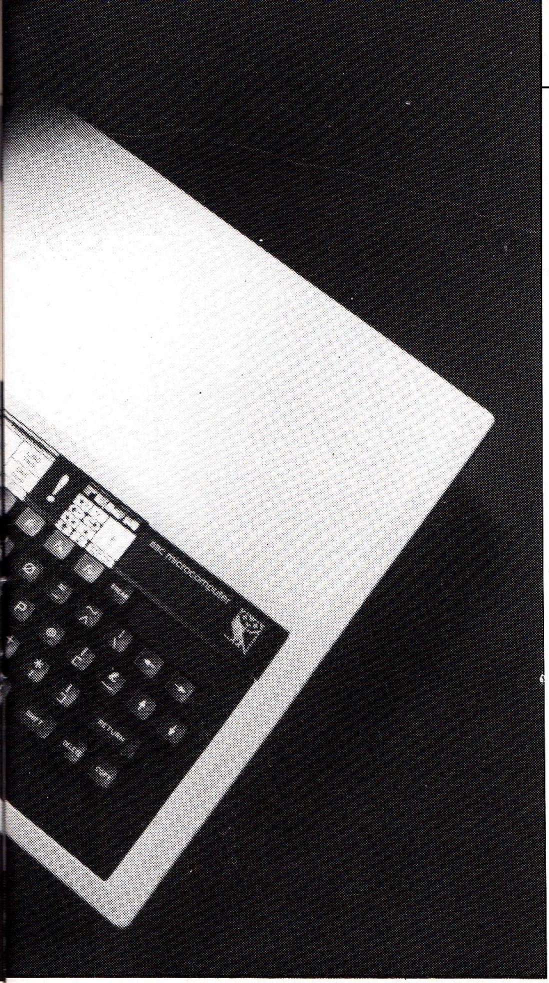
Microdrives should not be ignored, however – the first units to come on the market were in some cases unreliable, but newer machines are faster and reliability no longer seems to be a problem. Incidentally, asked about the plight of owners of early machines, Sir Clive recently said that any QL owner who was unhappy with the reliability of the microdrives would have them brought up to scratch if they returned the machine.

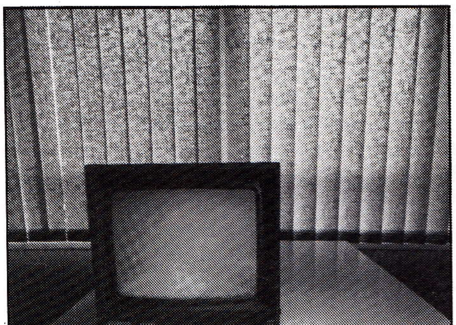
The BBC Micro itself is well catered for by third-party peripheral manufacturers. Disk units in the range of 100K to 800K capacity are readily available, though they don't come cheap. With separate RGB and composite video outlets, the BBC is easy to connect to monitors. For all-round excellence, an expensive RGB type is the only viable option, at over £300, but if like most users you only need colour when playing games, the most economical way to upgrade is with a monochrome moni-





The BBC B complete with key designation strip.
Apple's latest IIc, with second drive, mouse and display (below).
Amstrad's CPC464 and lone monitor (far right).





tor costing about £90. Use the TV for games. A monochrome monitor is inherently less tiring to look at, and even the very best colour displays will be hard pressed to produce text of equal quality.

Incidentally, the BBC's chronic memory shortage means that unless you have a Z80 or 6502 second processor you won't be looking at an 80 column display very often. Most software (including word processors) is configured to use a 40 column display as standard.

The Amstrad's remarkable price includes a colour RGB monitor in the £349 package, or a monochrome alternative in the £249 deal. Since the CPC464 keeps its video RAM in a separate cache, the 80 × 24 display vital for CP/M software is available without any drawbacks. Unfortunately, the low cost has to be reflected somewhere and the colour display isn't really up to the mark for prolonged use in this mode. You would be better off saving £100 by going for the green screen which is much sharper (though obviously less satisfactory for games).

The money can be put towards the disk drive, which is a rather strange unit. It uses 3″ disks of a type developed by Hitachi, but which is losing out to the rival Sony 3½″ version in the popularity stakes. The disks themselves are double sided, but the presence of only one set of heads in the drive means that to access the other side of the disk, you have to take it out and turn it over. All Amstrad owners have tape storage built into the right hand side of the keyboard, just where you get the microdrives on a QL.

Like the Sony disks (and unlike the old-fashioned 5¼″ cardboard type), the 3″ wonders are equipped with tough plastic jackets which make them robust enough for a domestic environment. The disks are good for 200K apiece, though a little disappointing considering how modern the design is.

Apple's little IIc (well, it's a lot smaller than an ordinary Apple II) has a 5¼″ drive cunningly built into the side of the machine, where you could be forgiven for overlooking it. Capacity is not the best, at 140K, but most Apple II software is designed to be economical on disk space.

The screen display is switchable between 40 and 80 columns of text, and even using the special IIc monitor the rather crude matrix on which the 80-column characters are formed makes the display tiring after a couple of hours.

The IIc keyboard is especially attractive despite its compact size. The keyswitches are of very high quality, whereas the Amstrad, QL and BBC all vary from machine to machine, and have been known to stick.

The Mac is a rather different kettle of fish. Its single built-in 3½″ drive is a bit marginal for a machine at this level of sophistication, and the second (external) drive should be regarded as a must.

The display is, of course, wonderful. The word processor supplied, Macwrite, accurately reflects on the screen not only the layout and line spacing of the final printed text, but also its exact size and typeface, together with any special attributes such as underlining or italicisation.

Not everyone is wildly enthusiastic about the keyboard – it is all right as far as it goes, with good feel, but the lack of cursor keys annoys many people. You are supposed to use the mouse to move the cursor, but this is unnecessarily fiddly and time-consuming if all you want to do is delete one particular character a couple or words back in the document.

The Apricot F1 is straightforward in its choice of processor and operating system, but it is defiantly original in physical design. The keyboard is, sadly, a retrograde step from that of the earlier Apricots, and ressembles that of the QL. It is completely flat and tends to be obstructive unless the keys are pressed with a careful vertical prod. It is also a remote unit, linked not by the curly cord of the Mac or Sanyo but by an infra-red link similar to that of a TV remote control. This sounds good, but in practice it is less than wonderful. You have to point the keyboard fairly accurately at the receiver on the computer, and if anything gets in the way it can take a while to realise your keystrokes are going astray.

To get around these problems you can use an optic fibre cable to link the two, but since this isn't as convenient as the original coiled wire cable the whole thing seems like a waste of effort.

Using the special Apricot monochrome monitor you get a very clear text display, but one which is rather small. Disk storage is by a single 3½″ Sony tape drive, offering an impressive 720K of storage per disk.

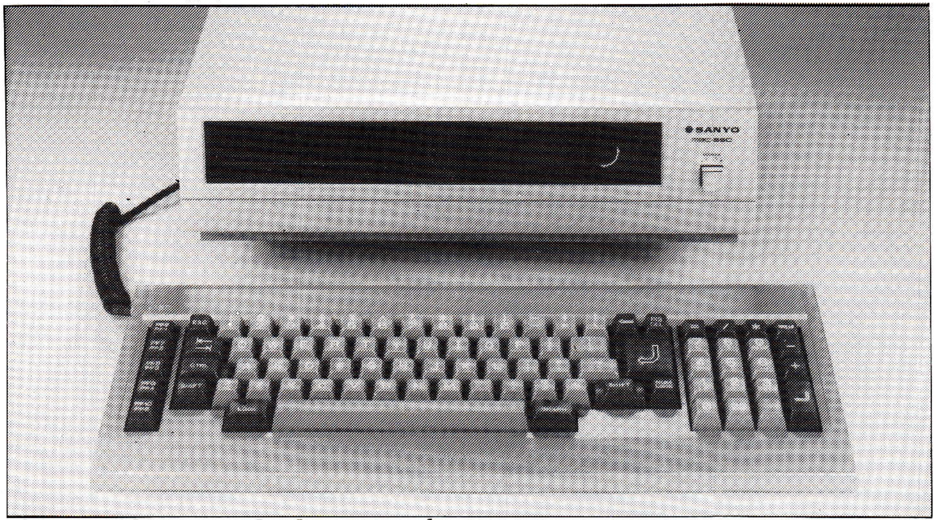
The Sanyo is very conventional. The keyboard and display are obviously modelled with the IBM PC in mind, and the conventional 5¼″ disk drives (one or two according to budget) can give up to 400K per drive – there are various options. We would always recommend the highest capacity drives you can afford – there is a considerable saving to be had in buying floppy disks, and the increase in convenience is dramatic.

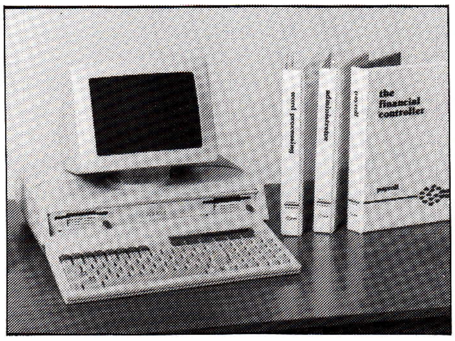
Seriously Though . . .

For use in the traditional home computer role, the BBC Micro and Amstrad have clear advantages over the other machines. The BBC's dominance in schools is still a force to be reckoned with, as is the completeness of the low-cost Amstrad package. Both machines are well supplied with high quality games, though the BBC is ahead in this respect. The QL is disappointingly short of entertainment software, so much so that even the relatively expensive Macintosh is far ahead here. Don't expect to have too much fun with an Apricot or a Sanyo – the only MS-DOS machine to be really well supplied with games software is the IBM PC.

If you want a computer for serious professional use at home, like many

▶





The much-publicised Apple Macintosh (right). Sanyo's MBC550 personal computer (just below). The Apricot (below right).

self-employed people, the BBC and Amstrad lose out. The BBC can be expanded to use a Z80 second processor with big-time disk storage under CP/M 2.2, but this costs big money. If your requirements are met by the four main categories of general purpose business program (word processing, spreadsheet, database and business graphics) then the high quality of the free QL suite makes that machine a much better bet for your £400. This is especially true when you realise that disks for the BBC will cost a good £250 minimum, and you might be able to make do with microdrives.

A better case can be made out for the Apple IIc. Through veteran Apple suppliers such as P&P Micro Supplies, a wide range of software can be obtained, although the newest and most powerful business programs need more processing horsepower than the ageing 6502 can deliver.

The Macintosh is something of an enigma. It is rather expensive in this country, but remember that the monitor is thrown in and a free (though rather basic) word processor is supplied.

The Mac's greatest advantage is its ability to draw and write in many typestyles. If you work with graphics at all, it is the natural choice, but it is restricted in the mainstream business applications.

The Apricot and Sanyo are the conservative choice for the professional user. Both run a wide range of software similar to that of the IBM PC, and the Sanyo can even interchange data disks (sometimes simple programs as well) with the IBM. The Apricot is a more elegant design, but the Sanyo inspires more confidence as a day-in-day-out computer suitable for heavy sustained use.

It is the long-term reliability question that, we suspect, puts many potential buyers off the QL. If it packs up, the only real way to get it fixed is via Sinclair Research, and this could leave you without a machine for some time. On the other hand, the present writer has an unenviable record with computers, blowing up BBC Micros, IBM PCs, Apple IIs, Commodore 64s, Spectrums *et al* with monotonous regularity, but he hasn't had a QL go down in flames yet.

If you are a programmer, you'll have your own ideas about what you want to program. Suffice it to say that there is nothing to touch the brute power of the QL for the money, and a good range of programming languages are coming on to the market.

The worst programmer's machine is, ironically, the one which offers the greatest challenges – the Macintosh. Apple is almost paranoid about releasing technical details, the high level languages are few and far between, and the only viable Mac development system (the Lisa 2, *aka* Macintosh Executive) has recently been given the chop. There is a big potential market for Mac software, but we can't see programmers rushing to meet the demand.

The Sanyo and Apricot are worthy engines for the programmer with an eye on the vast MS-DOS user base, but once again, we would opt for the Sanyo. It is more of a plain, ordinary computer with a very standard specification and operating system – the sort of thing you want if your programs are to be easily transferrable from one machine to another.

Considering that the QL is so young, it is doing remarkably well. It seems to be finding a niche in the home professional market and software now appearing supports it well in that role. Many would argue that its long-term success depends upon it remaining distinctive, apart from the mainstream business computer world. You can't beat IBM at its own game, so why bother trying? For this reason it will be a pity if too much emphasis is laid upon turning the QL into something it isn't, like adding a Z80 and 64K RAM to run CP/M. If you want a CP/M machine, buy one, they can be had cheaply if you know where to look.

It is a characteristic of the most successful personal computers that they eventually startle their own designers with the things they can be made to do. Classic examples are the Apple II, IBM PC, Spectrum and Commodore 64. We are nowhere near that stage with the QL yet – indeed perhaps the only program we've seen which really stretches its resources is Psion Chess. Let's hope that the software developers concentrate on moving forward, with really original programs which will further distinguish the QL from its rivals.

AN IMPORTANT ANNOUNCEMENT TO ALL QL OWNERS

bad or changed medium

Your data is worth a lot to you. Source programs, text, scientific information, records, all represent hours of your time and effort. Losing a file could at worst cost you money—and it's **always** infuriating!

No storage medium is 100% reliable. That's why TALENT has developed the CARTRIDGE DOCTOR. It's a sophisticated machine code program which will, in most cases, enable you to:

- recover files from a bad medium
- recover files which have been accidentally deleted
- recover files with lost or damaged blocks using the 'block patch' utility.

It's very easy to operate and no knowledge of BASIC or machine code is required.

Can YOU afford to be without the CARTRIDGE DOCTOR?

£21.95
+ 50p postage & packing
QL is a registered trademark of Sinclair Research

Available only from:

TALENT
COMPUTER SYSTEMS
Curran Building,
101 St. James Road, Glasgow G4 0NS
Tel: 041-552 2128 (ACCESS & VISA accepted)
SOFTWARE FROM SCOTLAND

NOW, buy a CARTRIDGE TIDY and you're a free! member of

- Holds 16 Microdrive Cartridges
- Interlocking design feature allows Horizontal and Vertical Stacking –Space Saver

CARTRIDGE CADDY SYSTEM

What Cartridge Caddy membership offers you

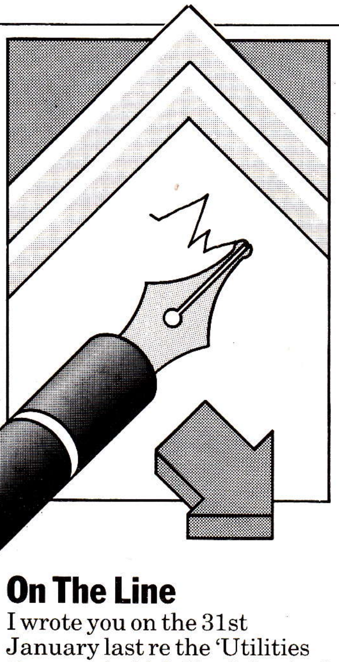
Microdrive cartridges at £1.50 delivered to your door! within 24 hours

Please send me . . . . (Qty) Cartridge Tidy's, □ZX □QL, at £6.99 each, inclusive. and . . . . blank microdrive cartridges at £1.50 each I enclose a cheque for £ . . . .

Make cheques payable to:
4 Systems, 4QL House, 68 Foxwood Close, Feltham, Middlesex
TW13 7DL Telephone 01–844 1399

Mr/Mrs/Miss
Address

4 Systems
68 FOXWOOD CLOSE, FELTHAM, MIDDLESEX TW13 7DL TEL: 01-844 1399



OPEN CHANNEL

This is the spot where we turn the magazine over to you, our readers. We welcome any comments, criticisms or anecdotes about either the QL or QL User. The address to send your letters is:

Open Channel, QL User, Priory Court, 30-32 Farringdon Lane, EC1R 3AU.

On The Line

I wrote you on the 31st January last re the 'Utilities Program' which I had checked and counterchecked, but, when asked to run, produced the legend – 510 error in expression.

Now however I have another problem, the 'Pacman' program, which I have also typed in and saved but when asked to run get 'line 100 bad name'. If that line is then deleted I get 'line 150 bad name' so, is it the programs or is it my QL? Please advise.

R Harrison
Skelmersdale

Without exception all programs published are first run on our QLs to make sure that there are NO syntax errors or fatal bugs. Immediately after this a listing is printed out and reproduced for publication without any modifications. This means if any errors are encountered we can confidently assume that they are not of our own making.

Furthermore, as the QLs in the office are standard JM versions the root of most problems has to be a typing error on the part of the reader. Aware that after many hours scrutinising listings and punching keys this conclusion may be difficult to accept, the only thing we can do is outline a couple of useful ways of spotting your own errors.

1. Error in expression – If a program crashes at a specified line with this message, immediately print out the values of all variables appearing on that line. If an asterisk appears instead of a value then you will have identified the offending variable and all that remains is to trace back through the program to where you have omitted to assign a value. If, as in the case of line 510, the variables constitute part of a procedure call, examine each element (separated by commas) in every call ensuring that the total number matches up (ie, compare line 490 with 780, 790, 710, 240 etc).

2. Bad name – Here the QL interprets a statement as a call to a non-existent procedure. If a procedure is indeed being called (instead of a keyword mispelt), check that the name by which it was defined matches up – (ie, in Pacman compare line 100 with 1120 and 150 with 1550).

3. Finally, everybody should keep a wary eye on punctuation marks and strange symbols (;, : & $£ /), as these are easily confused. Also, those with additional device drivers such as disks need to check carefully that any new keywords such as GET and PUT do not correspond to procedure names appearing on programs written on unexpanded QLs.

Spike Call

In a previous edition of QL User in the news section you described "The Plug" made by Power International. Unlike the other articles on that page, you did not give the address. Please will you remedy this.

C R Mann
Epping

We have been inundated by similar requests for the "Spike Spoiler".

The elusive address is: Power International, 2A Isambard Brunel Rd, Portsmouth, PO1 2OU – 0705 756715.

Transatlantic Opinion

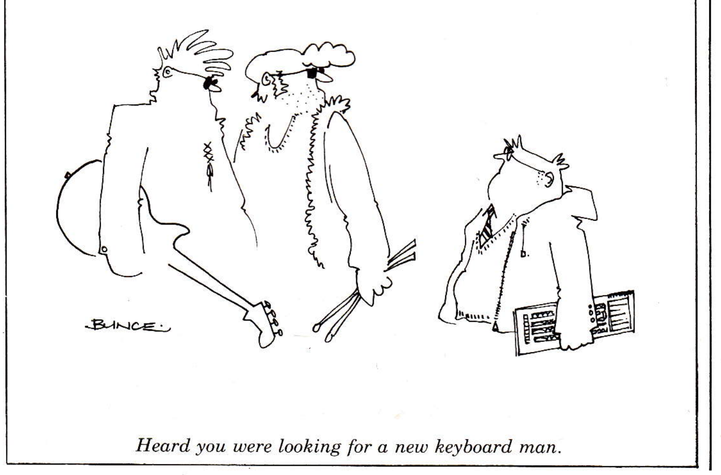
Notwithstanding my short acquaintance with the QL I'd certainly rate it as one of the more interesting small micros around. Turning the pages of the December issue, though, I discern many discordant voices. "SuperBasic is slow." Well, from the point of view of a microprocessor, *interpreting* a high level language throws a lot of emphasis on the speed of conditional jumps of branches. And, in this particular context, "32 bit addressing", as is used in the Motorola 68K inevitably tends to be a little cumbersome. To show its true colours, a 68K cries out for a *compiler*. With its extensive internal 32 bit register space, structured language support and capability for reasonably vast number-crunching without the addition of an expensive "co-processor" chip, even a 68008 can outclass say a Z80A or an 8088.

Another implied criticism is directed at the microdrives. It seems to me that for a price near to that of 1000 bit per second cassette recorders you get the ability to transfer large files at 100,000 bits per second, not much slower than many disk drives. And where else (US included) do you find a micro which has self-contained and reasonably quick access to large software or database packages, but can easily be carried in a small brief-case to wherever there's a TV. With time, surely we can expect that QL software writers will perfect the techniques necessary to let the drives be used in an efficient manner. And, perhaps, 200k or 400k cartridges for applications where slower access times can be tolerated? Elsewhere I see that the editing facilities for BASIC were judged to be less friendly than those of a Commodore machine. I think the reviewer should try editing a 'real' program, that is one that is more than one screen long, on each micro. For good measure, he might also set a variable or an array before the edit, and print it afterwards. I would be interested to know if his assessment remains the same. A QL-Apple Mac comparison would perhaps make even more interesting reading but that's another point (see elsewhere in this issue – Ed).

Finally I see a suggestion that a QL couldn't handle games or other programs where rapid screen movements are needed. Specifically the QL's large screen memory (32K versus 6.9K for a Spectrum) is cited. Well, as a test we might consider a loop to sequentially move the contents of a block of memory from location 1 to 2, zeroing location 1 to retore a blank screen. For a Spectrum, my best assembly language program would achieve this at a rate of 90 kilobytes/sec. For a QL, I estimate 400 kilobytes/sec. Whereas I concede this simple comparison may not tell the whole story, it seems that there is a fighting chance that the QLs speed can compensate for the larger screen file it has to handle.

In conclusion I think the QL is a very impressive machine for the price, and has considerable potential. Should a North American edition be offered (compatible with North American monitors), I for one would be interested.

P Wood
Ottawa



Heard you were looking for a new keyboard man.

Slow Boat

I noted your advice to Mr P Howland of Hereford in the March 1985 edition regarding the use of the F1 to F5 keys advising him to resort to Assembly language. He could alternatively access these keys via their codes, e.g.

```
100 CLS
110 a$ = INKEY $ (1000)
120 IF CODE (a$) = 232
    THEN CLS: PRINT "F1"
130 IF CODE (a$ = 236 THEN
    CLS: PRINT "F2"
140 IF CODE (a$) = THEN
    CLS: PRINT "F3"
150 IF CODE (a$) = 244
    THEN CLS: PRINT "F4"
160 IF CODE (a$) = 248
    THEN CLS: PRINT "F5"
170 GO TO 110
```

In practice of course the "CLS PRINT . . . etc" would be replaced by a call to a DEFined PROCedure.

Alec J Short
New Plymouth

This unfortunately is no alternative as it prevents all other keys on the QL's keyboard from being used to enter information. The effect Mr Howland was lookng for is to be found in 'The Progs' this month, where in effect a number of short programs can be hidden away to be run automatically whenever a particular function key is pressed. At all other times they remain invisible and do not interfere with whatever program happens to be running.

Running Wild

I read with interest the letter from Mr Benn in your February issue about using the EASEL screen dump routine, and decided to give it a try. After running Mr Benn's program I dutifully typed "scopy" and was rewarded with half a screen dump and an awful lot of junk.

What could be wrong? Mr Benn's routines did not seem to be at fault, and the screen dump worked perfectly when run from inside EASEL.

After considerable head-scratching I discovered that Psion's code seemed to be at fault after all, in that it did not seem to be completely position independent. My knowledge of QL machine code is very limited, but I thought that this was a real 'no-no'.

With my version of Qdos (AH), the RESPR( ) function put the code at location 261744 (3FE70H), but it would only run successfully when positioned at around 260720 (3FA70H), which means ranter a lot of wasted memory if it is the only routine you want. I do not have an assembler, so rewriting the code to be completely position independent seems a little daunting. Perhaps Mr Benn could suggest a solution to my problem.

Adam J Garstone
Sutton Coldfield

First Steps

I have just bought an assembler package for my QL hoping to start some serious programming. To my dismay I read the first page of the manual which says that it in no way attempts to teach 68000 or any other assembly language programming. I therefore now have, I assume, a perfectly good assembler but don't know what to do with it. Looking at all the book reviews any publication remotely connected with assembly language programming assumes that QL owners know all about it already.

Are there any books dealing with 68000 assembly language programming for beginners? Or do I have to buy a book for 'Spectrum' owners and convert at a later date?

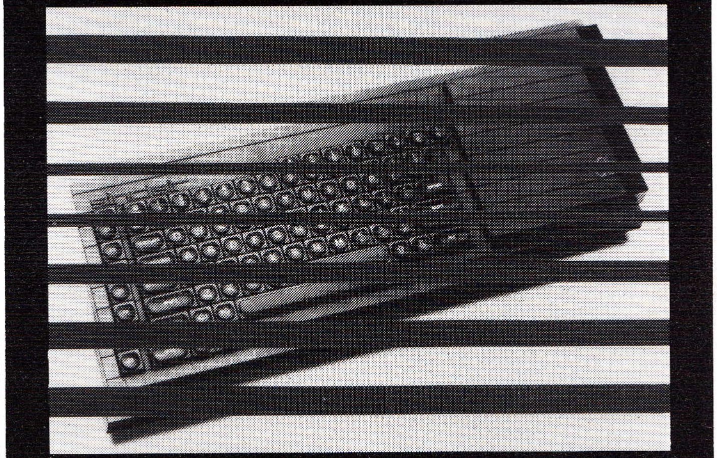
Also is there available, or have you heard rumours of anyone developing a light pen for the QL?

A G Weston
North Yorks

At present there is no single 68000 book suitable for the complete novice. However we recommend the following combination:
1. Osborne/McGraw-Hill's 68000 Assembly Language Programming by Kane, Hawkins and Leventhal to serve as your reference manual (£19.95).
2. The QL Advanced User Guide by Adrian Dickens from Adder Publishing which will provide you with all the relevant QDOS systems calls and subroutines (£14.45).
3. Adam Denning's Machine Code Tutorial, published in QL User, which as regular readers will know is available each and every month for the paltry sum of 95p(!)

NEXT MONTH

"THE EXPANDING QL"



In-depth review of the long-awaited multi-way expansion board and survey of peripherals it accommodates.

Software Review Special

Collected games and applications packages reviewed in a special feature.

Graphics Applications

Beginners' guide to the use of SuperBasic graphics commands – their potential and application.

Monochrome Monitors

The cheap alternative to the hi-res problem – black and white monitors for under £100.

Plus:

BROTHER EP44 AND TC600 COMMUNICATIONS

SOFTWARE APPLICATIONS FOR SMALL BUSINESSES

CONTENTS SUBJECT TO LATE REVISION

JUNE '85 EDITION ON SALE 21st MAY

NEW SYSTEMS PACKAGES FOR QL
NOW WITH
QDOS COMPATIBLE INTERFACE & DISK DRIVE

FRUSTRATED? DON'T BE. OUR SYSTEMS *DO* WORK

NO INTERFACE PROBLEMS	JUST PLUG IN AND GO!	FACTORY-TESTED BEFORE DESPATCH TO YOU	ALL CABLES SUPPLIED	PRINTERS SET UP READY TO USE	WE EVEN FIT MAINS PLUGS TO EVERYTHING

WE GIVE FULL TECHNICAL ADVICE *& AFTER SALES SUPPORT.* CAN YOU GET THIS KIND OF SERVICE *IN THE HIGH STREET?*

NEW — **CUSTOM SOFTWARE SERVICE** — **'TAILOR-MADE' FOR YOUR BUSINESS AT AFFORDABLE PRICES**

PRINTER & MONITOR PACKAGE DEALS + QL OPTION

PRINTER CHOICE (Standard) WITH	MICROVITEC COLOUR MONITOR	OR PHILIPS GREEN/AMBER MONITOR
Brother HRS + Mains Adaptor	£425	£265
Brother M1009	£459	£295
Epson RX80F/T+	£565	£420
Epson FX80	£725	£590
Epson FX100	£869	£735
Canon PW1080A NLQ	£649	£495
Daisystep 2000	£599	£445
Brother HR15	£715	£569
Epson DX100	£745	£590
Epson JX80 COLOUR (Superb with QL)	£895	

Printers are all supplied with RS232C interface. Monitors display full 85 columns of text.

WE MAKE IT ALL COMPATIBLE

'Help Sheet' supplied FREE with all printers.

SINCLAIR QL SUPPLIED WITH ANY SYSTEM — £389

NOW ADD DISK DRIVES!

Drive System including PSU & Computamate *'Q' Disk Interface*	**Single/Dual**	**Capacity (Formatted)**	**Price**
Teac 5.25" or 3.5"	Single	180K	PLEASE
Teac 5.25" or 3.5" 40/80	Single	360K	RING
Teac 5.25" or 3.5" 40/80	Single	720K	FOR
Teac 5.25"	Dual	2 × 180K	PACKAGE
Teac 5.25" 40/80	Dual	2 × 360K	DEAL
Teac 5.25" 40/80	Dual	2 × 720K	PRICING
'Q' Disk Interface alone			£149
Box 10 5.25" BASF SS/DD disks			£21
Printer cables for QL (Serial) Epson, Brother, Canon, Shinwa etc.			£9.20

MONITOR FOR QL
Factory set for full 85 column display

COLOUR	
Microvitec 1451DQ3	£275
MONOCHROME	
Philips	
BM7502 Green	£92
BM7522 Amber	£95
ACCESSORIES	
Epson RS232C I/F	£30
Epson RS232C 2K	£65
Miracle Systems I/F	£29.95

ALL PRICES INCLUDE VAT
FOR EXPRESS DELIVERY ADD £10 (SYSTEMS/MONITORS/DRIVES)
ACCESSORIES: P&P £2

ZEAL MARKETING LIMITED

VANGUARD TRADING ESTATE, STORFORTH LANE,
CHESTERFIELD, S40 2TZ
Telephone: 0246 208555 Telex: 547697

PSION PROBE

A new, regular questions and answers column where readers can interrogate Psion's Head of Software Support, Dr Irwin Joffe. Send your letters to: PSION PROBE, QL User, Priory Court, Farringdon Lane EC1R 3AU.

Processed Words

For some time now I have been investigating the idea of buying a microcomputer to use as a word processor.

I have been commissioned to write a series of educational monographs, of about 20,000 words each, and the quicker I can deliver the completed manuscripts the happier my publisher will be. Naturally. And me too.

I have been reading your publication for a while in the hope that I would find an overwhelming amount of evidence that what I need is the Sinclair QL with its Quill w/p program. I have not found this yet.

I would be most interested to know if any of your readers who regularly use Quill professionally, have any views on this matter.

I am particularly bothered about the keyboard, for instance, which I have tried, but not for extended periods. Also, things like slow screen handling, unreliability of the microdrive system and so on.

Can anybody advise me?

Rod McLoughlin
Rochdale, Lancs.

Both the QL and its Psion software have come a long way since they were first released, and most of the reviews written to date have been based upon the early systems. In particular, the QL software is now out in a new version which is significantly improved, and it is well worth your while to have another look at the whole QL package.

Printer Points

I am writing to you using the Quill program on my QL and a Smith Corona (D-100) dot matrix printer. It prints in single line spacing even though I have selected no line spacing in the 'Design' section of the commands. When listing a program the printer works very well. Enclosed with this letter is the list from a small program I have written which draws various patterns dependent on the numbers input into the program at the start. Could you please advise me on how I could print documents with no line spacing, how to print graphs from Easel and the patterns produced by the enclosed program as I cannot get any sort of print using the Quill printer program.

Sean J. Barnard
Jeddah, Saudi Arabia

The manual which accompanied your printer should contain a section dealing with how to set the DIP switches on the printer to allow it to operate using a whole range of different communication characteristics. In the description of the DIP switch settings, there will be one which deals with the way the printer responds when it receives an end-of-line or 'carriage return' character from its host computer. Look for CR/LF, CARRIAGE RETURN/LINE FEED, AUTOFEED, etc.

If the DIP switch is set to autofeed instead of CR/LF, and if you have set CARRIAGE RETURN and LINE FEED in the INSTALL_BAS program, then double-spaced lines will be printed. Try changing the DIP switch settings.

In order to print graphs from QL Easel, a graphics printer driver program is required for the type of printer you are using. QL Easel version-1 was supplied with a graphics printer driver for the Epson FX-80 only, whilst Version-2 has increased the range of compatibility of this printer driver, as well as adding graphics printer drivers for the Epson JX-80 colour dot matrix and the Integrex 132 ink-jet printer. Unfortunately the Smith Corona D-100 was not included in this upgrade.

En Coleur

I have a Sinclair QL and an order at the newsagent for QL User. I am only 14 years old. There are a few things I do not understand though, and wondered if you could help me out? I have tried putting programs onto a microdrive the way it says in the User Guide and all the QL responds with is NOT FOUND. Please could you give me instructions on how to put programs onto a microdrive. I think the microdrives might be faulty. And also, how would you write in colour, say 'Sinclair' in red, white, blue and black. It does not say how in the QL User Guide. I would be pleased if you could help me.

Arun Chada
Barrow on Soar, Leics

The QL will report the error message NOT FOUND if it is unsuccessful in finding a file which corresponds ***precisely*** *with the file name you have typed in. This means that in order to find a file, you need to type in the name of the file in exactly the same form as was typed in when it was first SAVEd. Either you are not doing this or it was not saved in the first instance.*

SuperBasic has the command INK to change the colour in which output to the screen will be written, and PAPER to select the background colour. In order to print QL USER in red on a black background, type:

```
PAPER 0
INK 2
CLS
PRINT "QL USER"
```

The ink and paper colours are documented in the 'Concepts' section of the QL Manual under 'Colour'.

Archive Aggro!

I thought I would avail myself of the 128K memory of the QL to create an index of LPs. Using Archive, I set up a file and started entering details. When less than one-twelfth of the way through, up popped an "out of memory" report. No more commands could be entered – so the data could not be accessed and my time had been wasted.

I approached Psion. Amongst the papers sent, I read that only 12K (yes, twelve) was available for file information – this is less than I can enter on my old ZX-81!

This negates the whole package until I have 500K available. Who on earth wants about a dozen separate files to access a few hundred LPs?

G L Budden
Burnham-on-Sea

It would seem that you have fundamentally misunderstood how QL Archive uses the QL's memory. The 12K which you refer to as being available for file information does NOT represent the total amount of storage available for data about your record collection. Rather, it is used by QL Archive to hold information about ***your information,*** *how this data is stored and in what sequence to present it if you want to use it in an ordered way. The actual data about your record collection is kept on a microdrive, and this has a capacity of about 100K.*

The 12K user area mentioned above is the amount of memory which is left over on the standard QL when the QL Archive program has been loaded, and is divided into two main parts. About 4K is used for the screen and as a buffer area for current records, and the remaining 8K is used to hold directories relating to the file organisation. For every record in the file, 6 bytes are used for building a basic index of the file, and a further 8 bytes per record are needed for each data field on which you order the file. Thus, for an unordered file, the index permits QL Archive to support a maximum of about 1,250 records (ie, 8K/6). If you order on one field, this maximum decreases to about 370 records, and so on.

Version 2 of QL Archive is more compact than Version 1 and leaves 21K free in the standard QL. This more than doubles the number of records it will support. Similarly, if you install a memory expansion in your QL, more memory will be available for the index and so the capacity of the database will be enlarged. Ultimately, the maximum size is limited by the overall capacity of the microdrive cartridge.

Power Problems

You recently carried a story about a device that claimed to eliminate the problem of spikes in the mains supply. I have certainly found that my QL occasionally 'freezes' for no apparent reason. (I use it principally for word processing with Quill.) But is there any reason to believe the QL to be more sensitive to spikes in the mains than other home computers? Nobody else seems to worry very much about it, and I have certainly never had this problem with my trusty old Vic 20.

On another matter affecting reliability, are you sure that the extra memory, such as offered by Quest would eliminate constant recourse to microdrives in using Quill? I understand that Quill Mark 2 will itself eliminate overlays, but does the presence of extra memory mean that it will no longer constantly save to an edit file on mdv2? It is here that many of the problems seem to arise.

Angus Ross
Norwich

Power supply spikes do seem to affect some computers more than others, and if the supply in your area is particularly bad, it might be a good idea for you to get a mains suppressor anyway. It might also be useful for you to check out your power supply to see whether or not this is the source of your problem... . Borrow a friend's power supply and see if this improves the reliability.

Regarding the effect of adding extra memory to your QL, this will generally be to speed up the operation of the Psion software packages, including QL Quill. It will also reduce the frequency with which the microdrives are accessed. This is because a large part of the extra memory is allocated to the microdrive buffer so that much more data can be typed in before reference need be made to the microdrive. For example, with a 64K RAM expansion you would be able to type in about 20 pages of a QL Quill (version 2) document before the QL uses the microdrive.

Stamp out microdrives...

...with the Computamate Q-Disk Controller

The only thing stopping your Sinclair QL from becoming a fully fledged business computer is the Microdrive. Your problems are now over . . . .

- Computamate Q-Disk immediately upgrades your QL to reliable disk storage (up to 1.4 MBytes).
- Computamate Q-Disk is available *now*: we guarantee dispatch within 48 hours.
- Computamate Q-Disk is the best QL Disk Drive System available, and has been highly acclaimed in press reviews — including QL user, March '85
- Computamate Q-Disk is completely Sinclair compatible, with software written by QDOS author, Tony Tebby.
- Computamate Q-Disk is fully guaranteed for two years.
- Computamate Q-Disk offers random access filing and Tony Tebby's Toolkit*, with over 30 extra commands on ROM. File finding time reduced by factor of 100.
- Computamate Q-Disk can be supplied with compatible disk drives, featuring the superlative TEAC mechanism, the best available.

*Expansion box (Q + 2) to be announced this month! This will allow expansion RAM to be used with your Q-Disk.

*Toolkit is a registered trademark of Q-Jump Ltd.

Q-Disk Controller
£149.00
(fully inclusive)

Disk Drive Systems from
£249.00
(Including Q-Disk, Mail order only)

Computamate

(Computamate Data Products is a division of Thomas French and Sons PLC)

MAIL ORDER
Computamate Data Products,
Scotia Road, Burslem,
Stoke-on-Trent. ST6 4DX
TELEPHONE ORDERS: (0782) 811711
Access/Barclaycard welcome

Q-DISK DEALERS

W.H. Smith & Sons — contact your nearest store
Strong Computers (0267) 231246 Wales
Eidersoft 01-478 1291 Ilford
(Eidersoft QL Art now available on disk)
Computers of Wigmore Street, 01-935 2452
Castlehurst 01-446 2280
Wanderbridge Ltd., 01-995 9060
Computer Enterprises Int., 01-542 4850
Printerland (0484) 514105 Huddersfield
Zeal Marketing (0246) 208555 Chesterfield
Compware (0270) 582301 Crewe
Slave Software (software for farmers)
050 846 8866 Norwich
Terry Blood Stores
(0782) 620321 — Trade only

Dealerships available.

Return to: Computamate Data Products, Scotia Road, Burslem, Stoke-on-Trent. ST6 4DX.

Please send me further details of the original, and the best QL Disk Controller available — The Computamate Q-Disk

Name ....

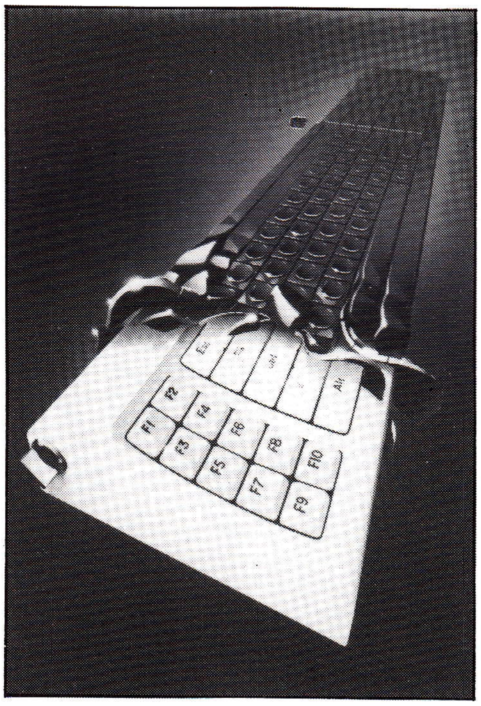
Address ....

Tel ....

The Drive for Excellence

TERMINAL EMULATION

The third part of our communications terminal by Adam Denning using BCPL and machine code.



Before continuing with this month's instalment, a reader has asked why the emulator program is written in BCPL rather than SuperBasic?

The capabilities of SuperBasic are enormous, and most of the program could be written in it, but all the good bits would require far more machine code extensions than the BCPL version here does. The program would also be noticeably slower and would offer lower performance at the higher Baud rates.

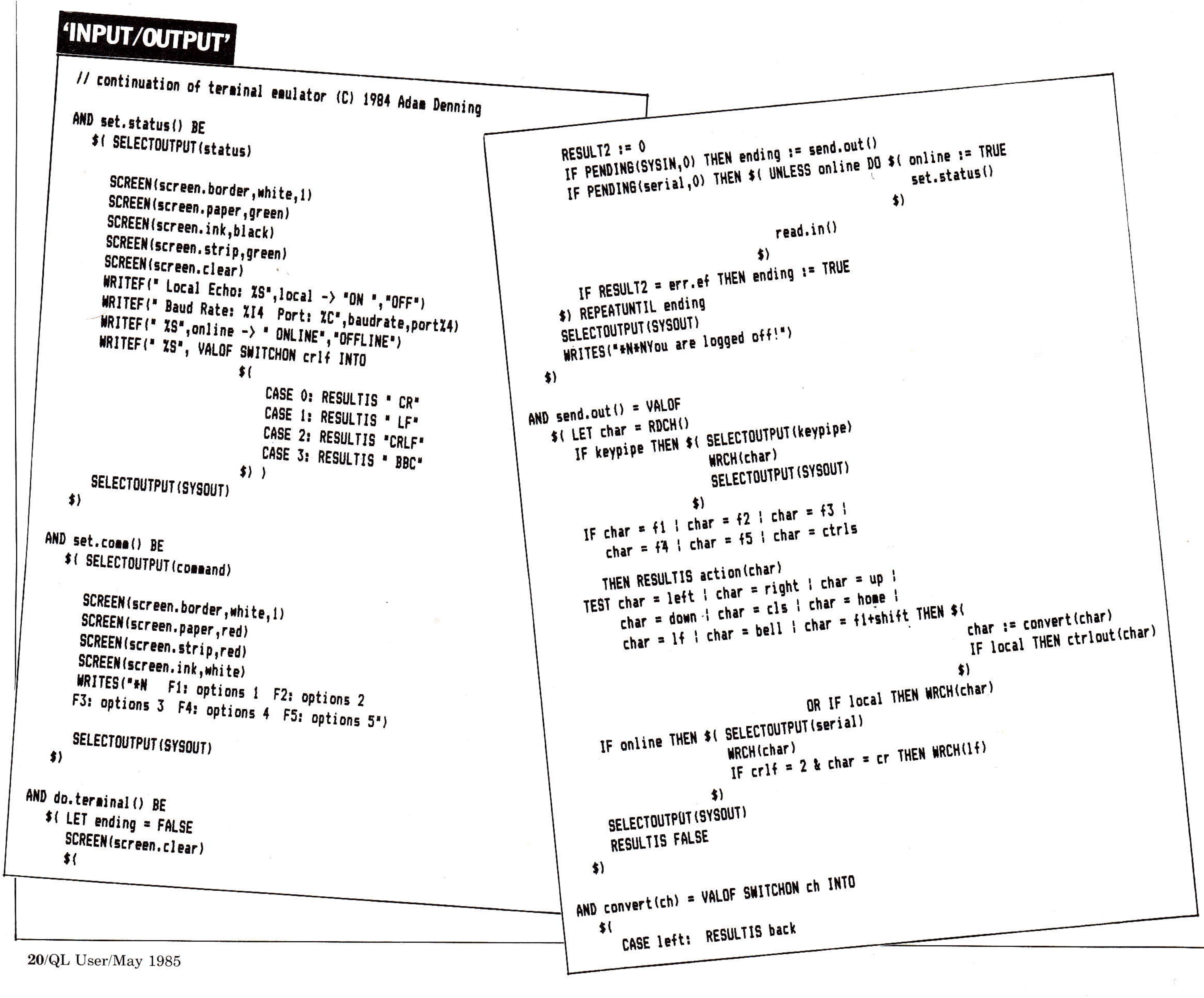
SuperBasic is unable to access all the more useful aspects of QDOS, such as the job control, the window size enquiry and the 'pending input' function. Extensions could be written to add these to the language but the program would be larger and you'd still need to buy an assembler. It wouldn't multi-task, either. That's why it's written in BCPL, although when Lattice C is finally released for the QL it could be re-written in that language.

The section of program shown this month has been carefully selected so that when combined with the earlier parts it will provide a (very basic) terminal emulation facility as it stands. The host of functions which the final program incorporates is well worth the wait! Before that, however, we've split the program into its constituent functions/procedures and they form the blocks below.

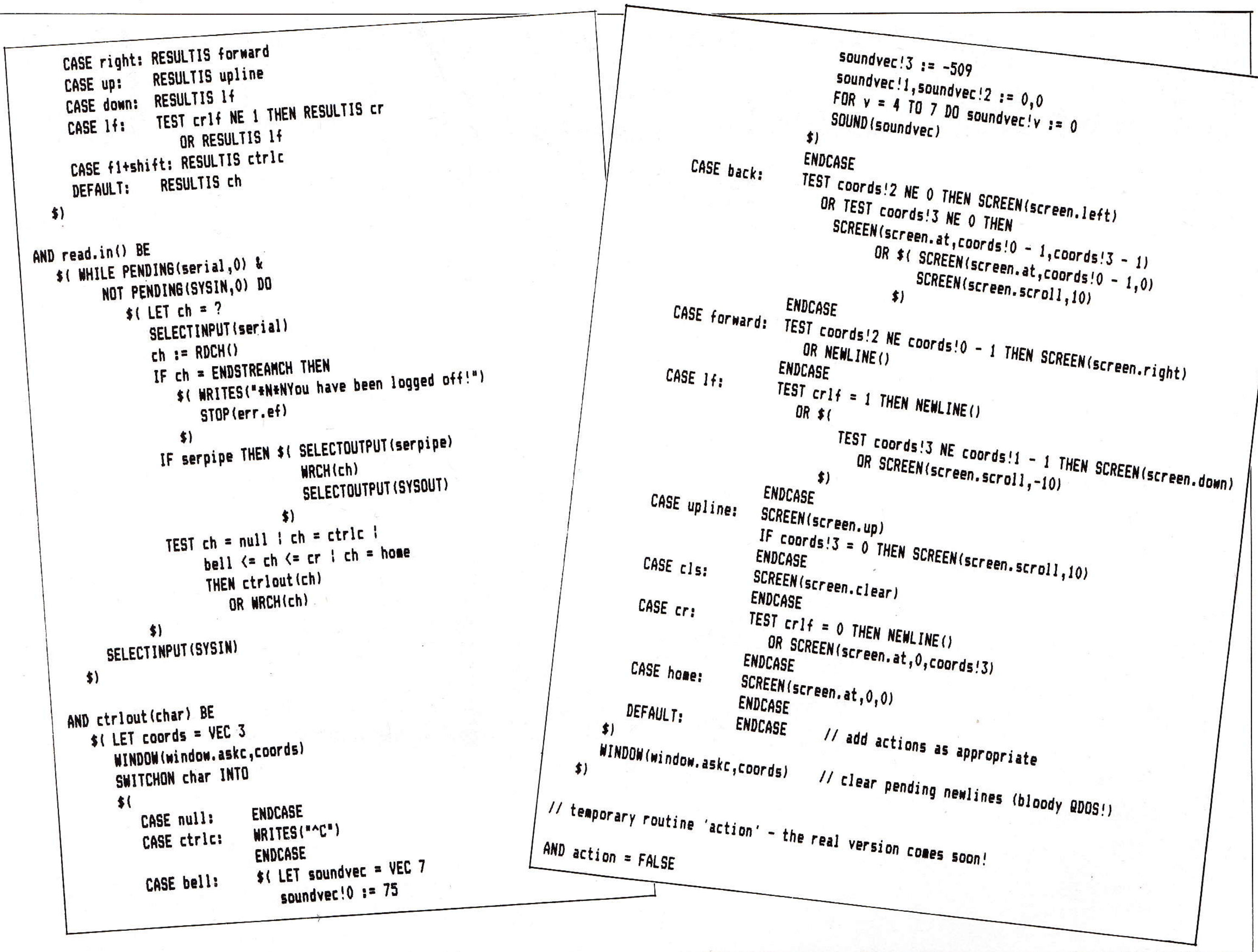
When this program section is combined with last month's, the whole lot can be compiled and linked with mc1_lib, mc2_lib, mc3_lib and mc4_lib to produce the first, elementary version of the emulator.

Next month we enhance this crude version and approach the finished emulator.

'INPUT/OUTPUT'



```
// continuation of terminal emulator (C) 1984 Adam Denning

AND set.status() BE
   $( SELECTOUTPUT(status)

      SCREEN(screen.border,white,1)
      SCREEN(screen.paper,green)
      SCREEN(screen.ink,black)
      SCREEN(screen.strip,green)
      SCREEN(screen.clear)
      WRITEF(" Local Echo: %S",local -> "ON ","OFF")
      WRITEF(" Baud Rate: %I4  Port: %C",baudrate,port%4)
      WRITEF(" %S",online -> " ONLINE","OFFLINE")
      WRITEF(" %S", VALOF SWITCHON crlf INTO
                     $(
                        CASE 0: RESULTIS " CR"
                        CASE 1: RESULTIS " LF"
                        CASE 2: RESULTIS "CRLF"
                        CASE 3: RESULTIS " BBC"
                     $) )
   SELECTOUTPUT(SYSOUT)
   $)

AND set.comm() BE
   $( SELECTOUTPUT(command)

      SCREEN(screen.border,white,1)
      SCREEN(screen.paper,red)
      SCREEN(screen.strip,red)
      SCREEN(screen.ink,white)
      WRITES("*N   F1: options 1  F2: options 2
     F3: options 3  F4: options 4  F5: options 5")

      SELECTOUTPUT(SYSOUT)
   $)

AND do.terminal() BE
   $( LET ending = FALSE
      SCREEN(screen.clear)
      $(
         RESULT2 := 0
         IF PENDING(SYSIN,0) THEN ending := send.out()
         IF PENDING(serial,0) THEN $( UNLESS online DO $( online := TRUE
                                                         set.status()
                                                      $)

                                      read.in()
                                   $)

           IF RESULT2 = err.ef THEN ending := TRUE
        $) REPEATUNTIL ending
        SELECTOUTPUT(SYSOUT)
        WRITES("*N*NYou are logged off!")
   $)

AND send.out() = VALOF
   $( LET char = RDCH()
      IF keypipe THEN $( SELECTOUTPUT(keypipe)
                         WRCH(char)
                         SELECTOUTPUT(SYSOUT)
                      $)
      IF char = f1 | char = f2 | char = f3 |
         char = f4 | char = f5 | char = ctrls

         THEN RESULTIS action(char)
      TEST char = left | char = right | char = up |
           char = down | char = cls | char = home |
           char = lf | char = bell | char = f1+shift THEN $(
                                                          char := convert(char)
                                                          IF local THEN ctrlout(char)
                                                         $)

                                     OR IF local THEN WRCH(char)

      IF online THEN $( SELECTOUTPUT(serial)
                        WRCH(char)
                        IF crlf = 2 & char = cr THEN WRCH(lf)
                     $)
      SELECTOUTPUT(SYSOUT)
      RESULTIS FALSE
   $)

AND convert(ch) = VALOF SWITCHON ch INTO
   $(
      CASE left:  RESULTIS back
```



```
     CASE right: RESULTIS forward
     CASE up:    RESULTIS upline
     CASE down:  RESULTIS lf
     CASE lf:    TEST crlf NE 1 THEN RESULTIS cr
                      OR RESULTIS lf
     CASE f1+shift: RESULTIS ctrlc
     DEFAULT:    RESULTIS ch
   $)

AND read.in() BE
   $( WHILE PENDING(serial,0) &
           NOT PENDING(SYSIN,0) DO
              $( LET ch = ?
                 SELECTINPUT(serial)
                 ch := RDCH()
                 IF ch = ENDSTREAMCH THEN
                    $( WRITES("*N*NYou have been logged off!")
                       STOP(err.ef)
                    $)
                 IF serpipe THEN $( SELECTOUTPUT(serpipe)
                                    WRCH(ch)
                                    SELECTOUTPUT(SYSOUT)
                                 $)
                 TEST ch = null | ch = ctrlc |
                      bell <= ch <= cr | ch = home
                      THEN ctrlout(ch)
                         OR WRCH(ch)
              $)
        SELECTINPUT(SYSIN)
   $)

AND ctrlout(char) BE
   $( LET coords = VEC 3
      WINDOW(window.askc,coords)
      SWITCHON char INTO
      $(
         CASE null:     ENDCASE
         CASE ctrlc:    WRITES("^C")
                        ENDCASE
         CASE bell:     $( LET soundvec = VEC 7
                           soundvec!0 := 75
                           soundvec!3 := -509
                           soundvec!1,soundvec!2 := 0,0
                           FOR v = 4 TO 7 DO soundvec!v := 0
                           SOUND(soundvec)
                        $)
                        ENDCASE
         CASE back:     TEST coords!2 NE 0 THEN SCREEN(screen.left)
                           OR TEST coords!3 NE 0 THEN
                              SCREEN(screen.at,coords!0 - 1,coords!3 - 1)
                                 OR $( SCREEN(screen.at,coords!0 - 1,0)
                                       SCREEN(screen.scroll,10)
                                    $)
                        ENDCASE
         CASE forward:  TEST coords!2 NE coords!0 - 1 THEN SCREEN(screen.right)
                           OR NEWLINE()
                        ENDCASE
         CASE lf:       TEST crlf = 1 THEN NEWLINE()
                           OR $(

                                 TEST coords!3 NE coords!1 - 1 THEN SCREEN(screen.down)
                                    OR SCREEN(screen.scroll,-10)
                              $)
                        ENDCASE
         CASE upline:   SCREEN(screen.up)
                        IF coords!3 = 0 THEN SCREEN(screen.scroll,10)
                        ENDCASE
         CASE cls:      SCREEN(screen.clear)
                        ENDCASE
         CASE cr:       TEST crlf = 0 THEN NEWLINE()
                           OR SCREEN(screen.at,0,coords!3)
                        ENDCASE
         CASE home:     SCREEN(screen.at,0,0)
                        ENDCASE
         DEFAULT:       ENDCASE      // add actions as appropriate
      $)
      WINDOW(window.askc,coords)     // clear pending newlines (bloody QDOS!)
   $)

// temporary routine 'action' - the real version comes soon!

AND action = FALSE
```

SET.STATUS
Looks after the green window at the top of the screen which is used to show the status of the terminal – the Baud rate, whether it's online or not, and so on. This is called the status window.

The procedure sets up the window's colours and then executes a series of WRITEF calls to print the current values. WRITEF is a library procedure which prints its arguments to the current output stream in the format specified by the first parameter, the 'format string'. A '%' character in this format string is treated as a special character, as the symbols following it tell WRITEF to substitute the whole composite ('%S' or whatever) with the relevant parameter. '%S' is replaced by a string, '%C' by a character and '%In' by a decimal integer using a field width of 'n'.

Notice how a number of the parameters to WRITEF are expressions here; maybe involving the '–>' conditional expression, maybe involving the value of a SWITCHON block, or maybe using the '%' byte indirection operator. This is one of the advantages of languages like this – it's very easy to select a particular value for a parameter by making it the result of an expression.

SET.COMM
Functions similar to set.status, but looks after the lower red 'command window' instead. This window was opened as a console device rather than a screen device as we will often need to read input from it. This does not affect any of the other windows that may be used.

DO.TERMINAL
This checks the status of the keyboard stream, and calls the send.out function if there is data waiting to be read. If not, it checks the status of the serial line, and calls read.in if there is data ready there.

Both read.in and send.out return results to do.terminal, the value usually being FALSE. If it is TRUE, then the program effectively ends.

SEND.OUT
This function deals with characters ready to be sent down the serial line. Having read the character and assigned it to 'char', the global variable 'keypipe' is checked. If this has a value other than 0 (FALSE), the keyboard pipe must be open, so the untouched character is sent down the pipe.

If the character is a function key (or CTRL-S) then the function to deal with special terminal routines is called.

If the key pressed is one of the cursor control keys or other 'control characters', it is first converted into a more general value before sending. Notice how we use SHIFT-F1 to send a CTRL-C character, as the key combination CTRL-C is trapped by QDOS and cannot otherwise be sent. At this stage, we also print the character down the serial line if the terminal is currently online (indicated by the global variable 'online' being TRUE).

CONVERT
This function takes an arbitrary control character and converts it to a more useful value.

READ.IN
This reads characters in from the serial line and takes appropriate action, such as displaying them on the screen. It does much the same as send.out but in reverse, and also checks to see if the terminal has been automatically logged off by the host sending a CTRL-Z (unlikely, but possible!)

CTRLOUT
This is the most complex function here as it has to convert control characters.

Pressing cursor-up usually moves the cursor upwards. It does this using a SWITCHON block to take a different action for each character value, and takes great pains to keep the screen in order. Characters like CTRL-C, NUL and BEL are handled easily enough, but the cursor movements are tediously difficult to achieve.

We first take a record of the current window size and cursor co-ordinates by making a call to WINDOW with the window.askc parameter. This returns four values, representing the X- and Y-co-ordinates of the window itself and the current cursor position, returning the values in terms of character positions rather than pixels (one of the things which SuperBasic can't do directly!) If the cursor is being moved left or right, we can generally use AT to move the cursor accordingly, but if the cursor is at the end of a line it must be put onto the line above or below. If this line is not on the screen, the entire screen has to be scrolled up or down by 10 pixels (the height of a character). Moving the cursor up or down has to follow the same scheme.

HARDWARE

Quest's Executive series aims to transform the QL into a versatile business machine — Paolo Baccanello follows their first attempts.

Quest International plc was the first major independent to commit itself to the QL. As far back as September 1984 the company had announced its intention to provide both hardware and software backup for the machine. On the one hand it promised a comprehensive range of peripherals which included floppy disk drives, Winchesters, RAM extensions and even an Expansion Console. On the other, it would offer CP/M 68K as an alternative to QDOS and support an extensive suite of accounting programs. The object was quite clearly to transform the QL into a fully fledged business machine available at a bargain price.

Coming at a time when bugs in QDOS and microdrive unreliability were undermining the QL's credibility, Quest's announcement did much to rally confidence in the machine. Only recently, however, have their units gone into volume production (with the exception of a Winchester) and even now certain aspects of their operation have yet to be finalised. The disk system reviewed this month is a case in point.

Quest's disk system is a radical departure from those reviewed to date. To begin with the interface board is very much larger measuring some 270 × 30 × 120mm and adding a full six inches to the QL's length. Housed in a matt black metal casing and resting on two plastic feet, (neither of which are angled) it is designed to fit unobtrusively into one of the four slots on their stylish aluminium Expansion Console (not available for review). Without this, however, slotting into the QL's main I/O port, it is a rather ungainly affair.

The interface's size is explained by the fact that it contains a special O/S card necessary if the QL is to run CP/M 68K. The 5¼″ drives it uses come in a variety of different formats affording capacities ranging from 200K up to 720K (formatted), dual or single drives. All are housed in high quality ribbed black plastic moulded cabinets measuring some 410 × 190 × 140mm. Though attractively styled, these tend to dwarf the QL. Half-height (double-width) versions are available. In either case the additional bulk has been set aside to include a fan, though this was not in evidence on the dual 720K drives reviewed. This was unfortunate as the disks did tend to get quite hot after extended use.

Despite minor niggles about their size the drives are well finished. In operation they purr indiscernibly and often flashing indicator lights are the only evidence of head movement. Furthermore the catches at the front of the machines may only be closed if a floppy disk or transport card is resident.

Firm Control

On the firmware side, Quest's disk controller is by far the most sophisticated of those yet reviewed. It supports 3″, 3½″, 5¼″ and 8″ drives though at present only the 5¼″

ON THE HORIZON

variety are in evidence. More importantly, it permits the user to run programs both under QDOS and Digital Research's CP/M 68K, though not simultaneously. Whilst the amount of software available under the latter system is by no means plentiful, until the current famine surrounding the QL ends, any alternative source of programs must surely be a boon. Incidentally, Quest's software arm, Padmede, currently supplies a comprehensive suite of accounting packages to run under this system. However, before moving into the CP/M 68K field business users should be aware that 80 track (720K) drives are not yet supported by it.

Double Standard

Conforming with the Sinclair-Tebby standard, Quest's QDOS driver permits disks to be formatted 512 bytes per sector, 9 sectors per track, 40 or 80 tracks. Indeed, we were able to interchange disks formatted on other systems with those working in this device. Reassuring proof that the standard is indeed effective!

However, having said this, sophistication here would appear to have bred complications. On the odd occasion, saving under Quill 2.00 crashed the program and corrupted the data disk. Unable to ascertain the exact source of the problem we assume that the driver provided (version 1.01) was probably pre-production. Certainly our reviewer, who had failed to make regular back-ups (*Silly boy – Ed!*) of the ½ megabyte of documents lost when the system went down for the first time, found it difficult to be objective!

In terms of extras, the emphasis in Quest's system is clearly on the CP/M side of the equation. Available on microdrive (O/S card extra) or disk this is a full implementation with all the regular commands and utilities as well as a 68000 assembler, C compiler and extensive manual.

On the QDOS side there are few frills. All you get is a disk driver, pure and simple. With others offering such goodies as extra SuperBasic commands, random access, multitasking control routines, disk doctors and even microdrive emulation, this seems somewhat meagre even with CP/M 68K in the balance. Nevertheless all existing QDOS I/O commands are supported and a necessary variant to the FORMAT command has been added. It takes the form:

FORMAT FDV <drive number>_<drive type>_<name>

and permits a maximum of three drives to be used (ie, FDV1, FDV2 and FDV3) and three different types of format to be specified (ie, 80 track double-sided, 40 track single or double sided). Also each disk may be given an arbitrary name (10 characters max) in much the same way as on microdrive cartridge. Finally it should be noted that the formatting program (QFORM) is separate from the driver and must be overlaid from either microdrive (mdv1) or disk (fdv1).

Similar to their RAM disk driver (reviewed last month) Quest's floppy driver (QFDL) comes on microdrive. Whilst this does mean that it is considerably easier for the user to update to new versions there are appreciable drawbacks.

First, the driver must be loaded in whenever the QL is switched on or reset. Loading takes twenty seconds, not unduly long but nevertheless a minor inconvenience when you consider that alternative ROM-based drivers are instantly accessible.

Secondly, the driver loads into RAM where instructions currently take four times longer to execute than those in ROM. The effect though slight is noticeable when loading or merging large files. On this score we understand that execution times have been halved with the latest version (1.02) of the driver.

The six page Disk Drive User Guide supplied with the system though short is succinct and to the point. There are plenty of warnings as to what might go wrong as well as useful suggestions such as backing up disks regularly, a golden rule well worth remembering.

Overall Quest's Executive Disk System is a bold and ambitious project. The company have taken an entirely different tack from the competition. In porting CP/M 68K across to the QL they have added a new dimension to the machine. Bearing in mind that this new operating system is available to only a few machines costing many times the price of a QL, this is no mean feat. However, its appeal to QL users will depend upon how much software support Quest can provide and at what price. At present they would seem to be catering for the upper end of the business market with the QL selling as a single integrated business system.

Prices:

Single drives		**Dual drives**	
200K	£295	2 × 200K	£469
400K	£419	2 × 400K	£579
800K	£499	2 × 800K	£695
Expansion Console			£139
Monitor Stand			£69.50
O/S Card			£49.50
CP/M 68K (disk)			£59.50
(mdv)			£99.50

Supplier: Quest International, School Lane, Chandlers Ford, Hants, SO5 3YY. Tel: 04215 66488.

Right Connections

SMC have added three products to their range. The first is a Centronics printer interface. Hardly a newcomer on the QL scene, it adopts the usual configuration with a lengthy serial cable at one end, short parallel cable at the other and interface located between the two. The device converts information sent at 9600 Baud and seems expensive bearing in mind that it is not switchable.

The second product is an adapter which allows most popular joysticks (9 pin Atari type) to be connected up to the QL. It consists of a short length of cable (235mm) with the appropriate connectors at either end.

The final offer is a 1.5 metre Epson printer cable. Now that Sinclair no longer provide one free with the QL this may well prove a winner.

Prices:

Printer Interface	£39.95
Printer Cable	£ 9.95
Joystick Adapter	£ 3.95

Stop Press

Unconfirmed reports say that the Medic Disk System will incorporate a Mac Emulator. Fact or fiction? We have yet to review any products from this source.

And with the demise of OE Ltd, users might try contacting Commpak Daka, 13 Beechwood Road, Uplands, Swansea – 0792 473697. The company produces multi-standard intelligent modems (MSM-PC) for the QL.

COMPUTER ONE MONITOR

Just another tool to help unravel machine code or a comprehensive debugging environment? Richard Cross suggests the answer lies somewhere in between.

A machine code monitor is a piece of software which helps in debugging a machine code program. The Computer One monitor is much more than this as it not only monitors machine code, but can monitor just about every aspect of the QL at a useful, down-to-earth low level.

The monitor is started by the Basic, EXEC command. Thus it runs as a job along with any others currently in the QL. Other jobs could easily be Computer One's editor and assembler, although with them all loaded there would not be much memory left in an unexpanded machine. With all three programs running you are in control of a very powerful machine code development tool. Source programs can be created with the editor converted to object code with the assembler, then the resultant program can be run/debugged with the monitor.

When the monitor is first loaded you are presented with five screen windows. The four main ones are for disassembling, dumping memory, dumping registers and inputting commands, each possessing its own channel number. These numbers are very similar to the SuperBasic channel numbers we are all used to. Specifying a channel number as one of the parameters for a command will cause output to be sent to that channel. New channels can be opened with the OPEN command to any device (eg, ser1, mdv1_). Each command has a default channel which will be used if no channel is specified. A very useful feature of the OPEN command is that if you open the CONsole (channel #1) to another device such as a microdrive file, then the monitor will take its commands from this file instead of the keyboard. So a set of commands which are used frequently could be stored as a microdrive file.

"The most impressive thing about the monitor is that any command can be run as a separate job using the CLONE command."

Typing HELP at the command prompt will cause a list of all of the commands the monitor understands to be printed. There are over 40 commands, each of which can be abbreviated to one or two letters. Machine code programs can be loaded by the use of the LOAD command. The program can then be set up as a job, but will be suspended. The job can be started by using the TRACE command. This causes a specified number of lines of the program to be disassembled and then executed. Also, after each instruction a register dump will be performed. Pressing the down cursor key will cause the TRACE command (or any other) to be repeated. Programs can be executed without tracing by using the GO command. This performs an unconditional jump to the current contents of the program counter.

"The Computer One monitor allows you to monitor just about every aspect of the QL at a useful, down-to-earth level."

The manual states that a register dump will be performed after a GO command finishes, this did not happen on my copy of the program. Execution of a GO command ends when a breakpoint is met and breakpoints can be set and cleared with the BREAK and NOBREAK commands.

However there appears to be an interesting point in the trace part of the monitor. If you try to end a program with an RTS instruction and trace through this instruction, the monitor may occasionally crash! The reason for this is that the monitor treats programs as jobs running under QDOS, these should be killed of explicitly or using the monitor. An RTS, therefore, is inappropriate. The QL can also be crashed by simply shifting a lump of memory over the system variables, or some other sensitive area, as no memory protection is performed. You should therefore be careful what you move to where.

Some of the other commands include moving and altering memory, searching memory, job management, heap management and channel management. A very useful one is WINDOW. This allows you to move a window about with the cursor keys and change its size with <CTRL> and the cursor keys. When <ENTER> is pressed the window size and position are printed in the window in the form 'SCR_256 × 100a0 × 0'. This is very valuable if lots of windows must be designed.

The monitor keeps an area of memory called the 'main' area. This is usually set to the start of a job. A pseudo register called BP (base pointer) will point to its bottom and TP to its top. All addresses will be assumed relative to this pointer unless a pound sign is given after the address. When the monitor prints any address, if it is within the 'main' area then it will be printed in the form disp(bp) eg, 01A4(bp).

One minor problem with the program is that once a disassembly or memory dump has been started it cannot be stopped until finished. This is a serious omission from the program, especially if you attempt to dump the entire contents of a large area of memory with a command like DUMP 0,10000 and change your mind half way through. You must then wait about half an hour until it finishes (when I did this I ended up pressing the reset button).

The most impressive thing about the monitor is the fact that any command can be run as a separate job with the CLONE command ie, the command will repeat on its own until you stop it. This is unbeatable if you need to keep an eye on specific areas.

The manual supplied is over 40 pages long and packed with all the required information. The first section, called 'getting started,' gives a demonstration of the monitor's abilities using the example program supplied on the cartridge. Overall this was good except for a few misprints – which had me lost for a few minutes.

The overriding impression of the monitor is of an invaluable part of any machine code programmer's tool kit. Apart from the few oddities mentioned, the program seems well worth the outlay.

Price: £24.95
Supplier: Computer One, Science Park, Milton Road, Cambridge.
(0223) 862616.

QL

SKETCH PAD £14.90 ★

PARALLEL PRINTER INTERFACE £34.90 ★

JOYSTICK ADAPTOR £ 5.90 ★

SKETCH PAD

A graphics design package allowing the user to create high resolution full colour pictures and diagrams using the cursor keys or a joystick. Features include:-

★ Screen dump to any Epson compatible printer.
★ Microdrive save and load of screens.
★ Use of stipple gives a vast range of colours.
★ Graphics and text can be freely mixed.
★ Automatic drawing of many shapes.
★ Filling of regular or irregular shapes in any colour.
★ Drawings saved on microdrive can be redisplayed from within any SuperBasic program using simple procedures supplied on microdrive.

Some applications include:-

★ Labels ★ Signs ★ Flow Charts
★ Graphics for Games
★ Artists Drawing

****

SPECIAL OFFER
SKETCHPAD PLUS JOYSTICK ADAPTOR £19.95

****

PARALLEL PRINTER INTERFACE

Simply plugs into the 'Ser I' port of the QL and any centronics compatible printer (eg:- EPSON, CANON, OKI, Kaga, Juki etc.) works with PSION software and requires no special commands. Full instructions included.

JOYSTICK ADAPTOR

Enables any Atari compatible joystick (eg: Kempston, Quickshot) to be used with the QL joystick ports.

All prices include V.A.T. and p & p within the U.K. Make cheques or postal orders payable to Sigma Research.

Sigma Research

231 Coldhams Lane, Cambridge CB1 3HY.
Telephone: (0223) 214109

CARE ELECTRONICS

UNIT 14, PEERGLOW INDUSTRIAL ESTATE, OLD'S APPROACH, TOLPITS LANE, WATFORD, HERTS WD1 8SR.
☎ Tel: 0923 777155 ☎

PRINTER INTERFACE
QL SERIAL TO PARALLEL CONVERTER

Switchable baud rate. This is necessary because the baud rate on the two serial ports cannot be set independently, and unless you can set the baud rate on serial to parallel converters your printer will be inoperative while using modems and other communications devices. Self-contained, colour matches the QL, full one year guarantee, comes complete with 1.5 metres of cable.

★ BAUD RATE SWITCHABLE
★ CRYSTAL CONTROLLED
★ QDOS and SUPERBASIC compatible
★ Drives any CENTRONICS printer
★ Price £49.91 (d) inc. VAT

Attention all software houses, Eprom cartridges for the QL, ROM expansion slot now available, an ideal way of packaging your utility program fast loading/no error and saves expense of microdrive cartridge. 1 cartridge £6.90 inc. VAT (c); 10 cartridges £5.29 each inc. VAT (c).
Please consult factory for details.

How to order by post, enclose cheque/P.O. made payable to Care Electronics or use Access.
7 days for delivery.
Please add carriage (a) = £8.00
(c) = £1.00
(d) = £2.00

Open
9am-5pm Mon-Thurs
9am-4pm Fri-Sat

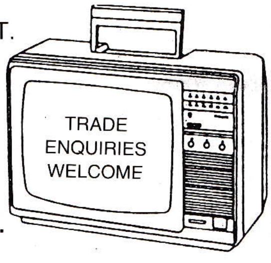
PRINTER SELECTOR

Capable of taking one QL and up to six printers
Price £75.90 (d) inc. VAT

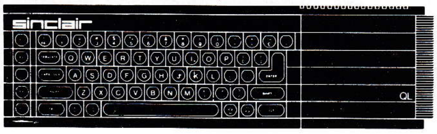
DON'T BUY A COLOUR MONITOR FOR YOUR QL

Have your Pye/Philips or Ferguson TX TV converted to a TV/Monitor.
In house conversion. £63.25 (a) inc. VAT.
Conversion kit £51.75 (d) inc. VAT

★ Full 85 Characters in monitor mode
★ Resolution better than 585 × 450 pixels
★ Image clarity comparable to leading monitors
★ Includes RGB lead for connecting with QL
★ Conversions carried out at our workshops within 2/3 days



Philips 14" TV/Monitor at £253 (a) inc. VAT.
Specially converted to run for the QL.



ADDER

QL Computer

68008 CPU, 128K RAM, 4 software packages	**£347** (a)

Monitors

Microvitec CUB 653 14" colour	**£239** (a)
Sinclair QL Monitor (Data Efficiency) colour	**£260** (a)

Printers

SC1000 dot matrix 100cps with RS232, Centronics, Fricton and tractor feed	**£189.95** (a)
SC1200 dot matrix 120cps printer with Centronics interface (requires ser to Centronics converter unit)	**£215.95** (a)
SC1200 as above plus Near Letter Quality print option	**£247.95** (a)
Serial to Centronics converter unit (plugs into ser Port)	**£24** (b)

Disc Drives and Interfaces

INTERFACE – supports up to 4 double sided, double density disc drives with microdrive emulation facility, useful utilities in ROM, disc editor and fully compatible with QDOS and PSION packages	**£99** (b)
DISC DRIVES – 80 track, double sided, double density 5.25" disc drives offering 720K per drive. Boxed with PSU and leads	
720K single drive	**£148** (a)
1.4Mbyte dual drive unit	**£304** (a)
Memorex 5.25" discs DSQD (box of 10)	**£38.95** (b)
80 track single sided, double density 5.25", 360K per drive	
360K single drive	**£130** (a)
720K dual drive unit	**£274** (a)
Memorex 5.25" discs SSQD (box of 10)	**£34.60** (b)
Library box (holds 10 discs)	**£1.95** (c)

Memory Expansion Cards

64K RAM	**£104** (b)
128K RAM	**£165** (b)
256K RAM	**£304** (b)
512K RAM	**£504** (b)

Business Package

Complete business package comprising a QL computer, Microvitec CUB 653 14" colour monitor, SC1000 printer and all connecting leads	**£769** (a)

Communications

Link your QL to the outside world via the telephone!

QCON RS232 interface control unit for QMOD and QCALL	**£69.95** (b)
QMOD modem 1200/75 and 1200/1200 baud for use with QCON	**£69.95** (b)
QCALL auto answer/dial unit for use with QCON and QMOD	**£43.95** (b)

Accessories

Microdrive cartridges 1–9 (each)	**£1.75** (c)
Microdrive cartridges 10+ (each)	**£1.65** (c)
Library box (holds 20 mdvs)	**£5.20** (c)
RS232 serial to Centronics converter unit	**£24** (b)

Language Software

PASCAL (Computer One)	**£34.50** (c)
LISP (Metacomco)	**£52.20** (c)
BCPL (Metacomco)	**£52.20** (c)
FORTH (Computer One)	**£34.50** (c)

The ADDER QL MACRO Assembler
£29.95 (c)

". . . the best editor/assembler going for a beginner or experienced user . . . it is the most 'user-friendly' editor/assembler I've ever used!" (Quanta, Jan 1985)

The ADDER MACRO Assember provides all the facilities required to develop and debug 68000 assembly language programs. The integrated editor/assembler resides in memory, eliminating the necessity to assemble from slow microdrives (large files can of course be assembled from disc or microdrive if required). All standard Motorola compatible features are provided by this assembler, plus the ability to change registers and memory for debugging purposes.

Other Assemblers

Metacomco QL Assembler	**£34.75** (c)
GST (Sinclair) QL Assembler	**£34.75** (c)

Utilities

Monitor/disassembler (Computer One)	**£21.95** (c)
QL Toolkit (Sinclair)	**£21.95** (c)
QDOCTOR (ADDER)	**£14.95** (c)

STOP PRESS . . .
NEW PRODUCT

NEW!

Ever lost a program stored on microdrive?
You need a microdrive doctor. QDOCTOR provides a full screen microdrive sector editor/checker which checks a complete cartridge for errors, then recovers as much data as remains intact. Have you ever deleted a file by mistake? Recover it with QDOCTOR! No user of the QL can afford to be without this valuable utility. Comprehensive manual included. Only **£14.95** (c)

Business and Applications Software

QL Typing Tutor (Computer One)	**£17.35** (c)
Cash Trader (Sinclair)	**£21.95** (c)
QL Entrepreneur (Sinclair)	**£34.75** (c)
QL Project Planner (Sinclair)	**£34.75** (c)
QL Decision Maker (Sinclair)	**£34.75** (c)

Games

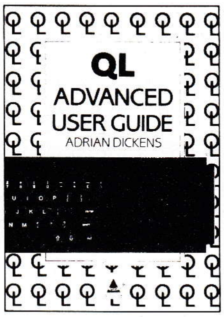
QL Chess (Psion)	**£17.35** (c)
WEST Adventure (Talent)	**£17.35** (c)
ZKUL Adventure (Talent)	**£17.35** (c)

Connection leads – two metres long

ser2 – RS232 lead	**£11.95** (c)
ser2 – ser1 (QL to QL)	**£5.95** (c)
ser2 – BBC Micro RS423	**£5.95** (c)
QL — monochrome monitor lead	**£5.95** (c)
QL – Microvitec colour monitor lead	**£6.95** (c)
SER port connector + 2m of wire	**£3.95** (c)
CTL port connector + 2m of wire	**£3.95** (c)
CTL port – Joystick (ATARI standard) lead	**£5.95** (c)

Books (No VAT)

Advanced Prog Sinclair QL	**£6.95** (b)
Artificial Intelligence QL	**£6.95** (b)
Assembly Language Programming QL	**£7.95** (b)
Desk-Top Computing with Sinclair QL	**£6.95** (b)
Developing Applications on QL	**£6.95** (b)
Good Prog with QL Super BASIC	**£5.95** (b)
Intro to Simulation Techniques QL	**£6.95** (b)
Intro to The Sinclair QL	**£6.95** (b)
Intro to Super BASIC with QL	**£6.95** (b)
Quantum Theory (Century)	**£5.95** (b)
QL Abacus (Century)	**£8.95** (b)
QL Advanced User Guide (ADDER)	book **£12.95** (b)
QL Advanced User Guide Programs (ADDER)	cart **£9.95** (+VAT) (c)



"An excellent reference book . . . a stepping stone to the heart of the QL." Sinclair User, January 1985.

QL Assembly Lang Prog (McGrawHill)	**£12.95** (b)
QL Easel	**£7.95** (b)
QL Super BASIC (A Prog's Intro)	**£7.95** (b)
QL Games Book	**£6.95** (b)
QL Quill (Century)	**£7.95** (b)
Quill, Easel, Archive, Abacus	**£6.95** (b)
Sinclair QL Adventures (Sunshine)	**£5.95** (b)
Sinclair QL User Guide	**£7.50** (b)
Sinclair QL Companion	**£6.95** (b)
Word Processing with Sinclair QL	**£6.95** (b)
68000 Assembly Lang Programming	**£18.95** (b)
68000 Handbook	**£11.50** (b)

All prices exclude VAT.
Please add carriage (a) £8.00 (Securicor); (b) £1.50; (c) £1.00; and VAT at 15% to order value including carriage.

ADDER

MAIL ORDER ONLY

Please make cheque/postal order payable to:

ADDER Publishing Limited,
Dept QLU,
PO Box 148,
Cambridge CB1 2EQ
Telephone (0223) 277050

Official orders from Government establishments, universities, colleges, state schools and nationally recognised PLCs welcome.

Prices correct at time of going to press, but subject to change without notice.

Sinclair, QL, QDOS are trademarks of Sinclair Research Ltd.

AI ON THE QL

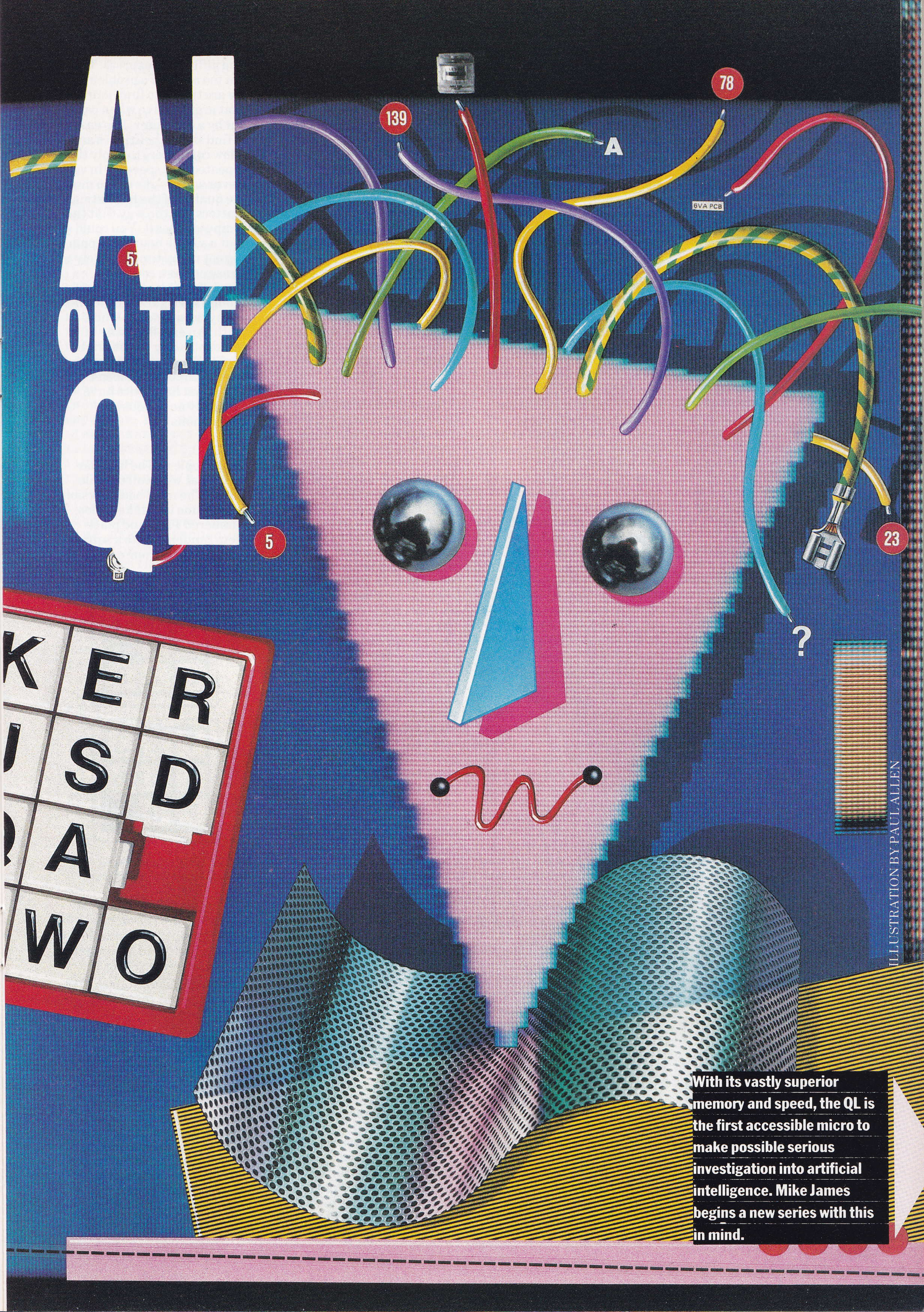


ILLUSTRATION BY PAUL ALLEN

With its vastly superior memory and speed, the QL is the first accessible micro to make possible serious investigation into artificial intelligence. Mike James begins a new series with this in mind.

AI ON THE QL

This short series deals with two exciting subjects – Artificial Intelligence (AI) and its practical QL implementation. AI is an important and interesting subject not only because it's most probably where the future of computing lies, but because it can supply techniques for using today's computers in new ways. The connection between the QL and AI is that the QL is one of the few machines that is powerful enough to run AI-based software; although it's true that for the larger programs it would still be necessary to use either machine code or a compiled language to gain speed.

Most introductions to AI are largely theoretical discussions of how things can or could be done. In the case of the QL, however, SuperBasic makes it possible to give practical examples of all the methods described in this series. Every program included will be as 'well structured' as possible and make use of all SuperBasic's advanced facilities. So, even if you are not particularly interested in AI you should find the programs useful as examples.

Artifical Intelligence is a very large field of endeavour, but we'll be concentrating on the two most practically important areas — problem solving/reasoning and natural language processing.

The first of these, problem solving, is a vast topic covering applications in game playing, expert systems, logic, etc. Natural language processing encompasses efforts to make computers, talk, understand what is said to them, translate between different languages etc. This month we concentrate on ways of getting computers to help with the solution of problems that are normally thought to require skills only humans possess.

Problems, Problems

To be able to write a program that solves a problem it is necessary to know how *you* would go about solving it. Anyone who has even the slightest practical knowledge of computers knows that there is no way of getting an answer to a question simply by asking the machine – someone must have already 'programmed in' the route to the answer. The trouble is, there are many problems that humans don't know how to solve and naturally enough these are the very problems that we would like computers to work out. Often, humans will achieve a solution without any idea as to how they did it and this is where the problem lies. To be able to write a program to solve a problem, it is not only necessary to know how, but also to be able to describe how, you arrived at the solution. This is because the program that you are trying to write *is* a description of how you solved the problem!

One approach to AI claims that it doesn't matter if we don't know how to go about solving a specific problem, the important thing is to write programs that can solve general problems – in the same way that humans do. Following this line you could end up with a program that is a general problem solver but can't 'tell you' how it did it! Programs that behave in this way are often said to 'learn' the solution to a general problem. The idea sounds very promising but so far very little progress has been made with it. A more practical alternative is to press on with the attempt to write programs that solve specific problems even though the complete method of solution may not be known.

Humane Solutions

Consider the problem of winning at chess or draughts. There are some humans who solve this problem very well. They tend to win against most opponents, which suggests that they have a method of playing which is a good solution to the problem. If there weren't such experts then you might decide that there was no solution to the chess/draughts problem. In other words, you might assume that players made moves for a wide variety of reasons none of which had anything to do with the strategy of the game – the winner was a random choice.

The existence of people who perform consistently better than beginners and the internal feeling of 'working things out' when you are playing such a game, both suggest that humans *do* have a 'program' inside their heads foı solving the problems they present. If you try to explain, however, the way you play chess or draughts you're very likely to fail. You might be able to manage vague generalisations or very complicated justifications for particular moves but, if you try to write a program to play in your place, you will soon discover how poor your knowledge is.

All this leaves us in a rather confused state of affairs. There are many problems that humans solve that are very difficult if not impossible to reduce to a set of rules that guarantee a solution.

Honourable Heuristics

Traditionally a program is a list of instructions for giving a sure solution to a problem or reporting that no solution exists. Such a list of instructions is called an *Algorithm* and constitutes the core of computer science and programming to date. However, as already pointed out, there are many problems that we have not found algorithmic solutions for and if computer science is to develop we must make progress in these unexplored areas. Examining the way you solve problems often reveals that what you are doing is not using an algorithm but applying a loose collection of rules that 'seem' to work. For example, in chess you might hold to the rule 'control the middle of the board'. Now, while this and other rules like it cannot guarantee to find a solution (ie, produce a win) they make it more likely that you will get closer to one. A rule that tends to get you closer to a solution is known as a *heuristic* and while it might seem that a heuristic is a 'second class' algorithm this is far from the truth.

Heuristics may not be able to guarantee a solution, and they cannot tell you when a solution doesn't exist, but they can be used in a wide range of situations and when they do come up with a solution it's often in much less time than an algorithm would take for the same problem. The future of computer science and programming is almost certainly going to be more about heuristic and combined heuristic-algorithmic approaches to problem solving.

Finding a heuristic still seems like a very difficult task and perhaps we are not much better off than before. The sort of heuristics that humans use are often difficult to discover and difficult to express. This is not too much of a problem since we are trying to find heuristics that are effective when carried out by a computer. It is easier to find simple heuristics and allow computers to apply them repeatedly at speed, or in very clever ways — it's not so much the quality of the heuristic that matters, it's the way that the computer uses it. You could say that a simple heuristic applied a great many times is likely to be as good as a complex heuristic applied few times; though such generalisations are by no means the whole truth. An excellent way of demonstrating the heuristic approach is to apply it to a simple game (this doesn't imply that heuristics have currently no serious applications).

Shifting Squares

Most people will be familiar with what we'll call the 'tile game'. The most basic version of the game is to take 8 tiles, numbered 1 to 8 and place them randomly into a square pattern leaving one empty space *(Fig 1a)*.

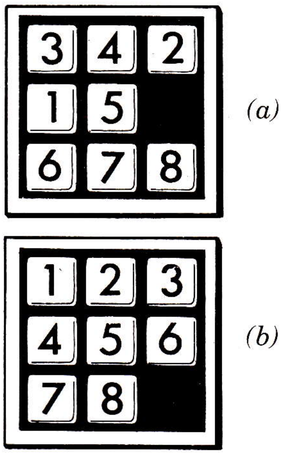


Figure 1. Initial (a) and final (b) positions for the tile shifting game.

The object of the game is to move tiles into the empty space, keeping the overall square shape, and arrive at the final arrangement *(Fig 1b)*.

A more modern version of the tile game is the familiar Rubik Cube (it's interesting to speculate upon a program to solve that along similar lines).

Take a look at the listing of the automatic tile program *(Fig 2)*. You should be able to see that it is made up of a number of procedures. Each one carries out a different part of the game and the program is a collection of procedures that can be used in different ways to explore the problem.

Before you can start to write any game-playing program you have to decide on how to

```
12    WINDOW 452,236,60,20
14    MODE 8
20    init
30    print_board
40    set_shuffle
50    find_space
60    REPeat game
70     win_pos
80     IF S%=0 THEN EXIT game
90     rnd_mve
100    do_move
110    MOVE%=MOVE%+1
120    print_board
130   END REPeat game
140   PRINT "SOLVED IN ";MOVE%
1000  DEFine PROCedure init
1010  DIM B%(9)
1020  DIM M$(9,4)
1030  DIM M%(9)
1040  Q%=0
1050  DATA "24  ","135 ","26  ","157 ","2468"
1060  DATA "359 ","48  ","579 ","68  "
1070  DATA 2,3,2,3,4,3,2,3,2
1080  RESTORE 1050
1090  FOR I=1 TO 9
1100   READ M$(I)
1110  END FOR I
1120  FOR I=1 TO 9
1130   READ M%(I)
1140  END FOR I
1150  MOVE%=0
1499  END DEFine init
1500  DEFine PROCedure print_board
1510  FOR I=1 TO 9
1520   PRINT B%(I);" ";
1530   IF I=INT(I/3)*3 THEN PRINT
1540  END FOR I
1550  PRINT
1560  pause_1
1999  END DEFine print_board
2000  DEFine PROCedure set_shuffle
2010  FOR I=1 TO 9
2020   B%(I)=I
2030  END FOR I
2040  P%=9
2050  N%=RND(25 TO 34)
2060  FOR Z=1 TO N%
2070   rnd_mve
2080   do_move
2090  END FOR Z
2100  Q%=0
2999  END DEFine set_shuffle
3000  DEFine PROCedure find_space
3010  FOR I=1 TO 9
3020   IF B%(I)=9 THEN P%=I
3030  END FOR I
3999  END DEFine find_space
4000  DEFine PROCedure win_pos
4010  S%=0
4020  FOR I=1 TO 9
4030   IF I<>B%(I) THEN S%=1
4040  END FOR I
4999  END DEFine win_pos
5000  DEFine PROCedure pause_1
5010  FOR I=1 TO 500
5020  END FOR I
5999  END DEFine pause_1
6000  DEFine PROCedure rnd_mve
6010  I%=RND(1 TO M%(P%))
6020  J%=M$(P%,I%)
6999  END DEFine rnd_mve
7000  DEFine PROCedure do_move
7010  T%=B%(P%)
7020  B%(P%)=B%(J%)
7030  B%(J%)=T%
7040  Q%=P%
7050  P%=J%
7999  END DEFine do_move
```

Figure 2. Listing of the prototype automatic tile game simulator (note that line 90 and lines 6000-6999 are deleted from the final version).

represent the 'board'. For the tile game it is easier to ignore the fact that the board is two dimensional and use the array M% with nine elements, one for each position that a tile can occupy. Each position is numbered so that when the final arrangement is reached tile one will be in M%(1), tile two in M%(2) and so on. The empty space can be represented by any convenient symbol but it makes life easier if it is a number and 9 is the most obvious choice. To find out which tile is at position 1, simply look at M%(1).

Using this representation, it is easy to print the board out (procedure print_board) and to check to see if the final winning position has been reached (procedure win_pos). It is not quite so easy, however, to set up a starting position. The problem is to arrange the tiles in a random order such that they could have been produced by legal moves of the tiles. If you just generate a completely unconstrained random order for the tiles you can produce a startling arrangement that *cannot* be moved to the final position. (The reason is because there are two versions of the tile game, a lefthanded version and a righthanded version and you cannot convert one to the other using only legal moves!) The solution is to set the board up in its final or target position and then 'scramble' it by way of a random sequence of legal moves. Procedure set_shuffle sets up a random board by shuffling a board previously set up in the final position. This has the additional advantage that the difficulty of the problem can be controlled by the degree of shuffling. The more the board is shuffled away from the final position the harder it should be to solve. (It might be interesting to test this proposition – Ed).

The final problem is to detect illegal moves, The quickest way to do this is to set up a table that lists all the possible moves for any position of the 'space' (the 9 in this representation). If you look at procedure *init* you will see that the string array M$ is initialised to contain lists of legal moves. For example if the 'space' is at position 3 on the board then looking at M$(3) gives the string '26' which should be taken to mean that you can move the tile in position 2 or the tile in position 6 into the 'space' but no other moves are legal. This easy representation of legal moves for any position of the 'space' is the main reason that the tile game can be programmed efficiently in SuperBasic.

The only other procedure that deserves mention at this stage is procedure do_move, which will make any move specified by P%, the current position of the 'space', and J%, the position of the tile to be moved into the 'space'. Obviously this is just a swap between M%(P) and M%(J) and the new position of the 'space' is J% which can be placed into P% to keep the current position updated.

Brains Or Brawn?

Using just these procedures it is possible to write a crude tile game program. Procedures inti, print_board, set_shuffle and find_space are called first. This part of the program is always the same and simply sets up the legal move table, initialises the board, and then prints it out. Procedure find_space has not been mentioned before, but its role is to locate the position of the 'space' and place it in P% after the board has been shuffled. This only has to be done once because the position of the 'space' is tracked by procedure do_move and is always in P%. Finding the solution begins with a check to discover if the solution has already been found by calling procedure win_pos. If it hasn't (ie, S%<>0), then the only thing to do is make a move. In this first simple program the move to be made is picked at random by procedure rnd_mve. This is acting as a 'move generator'. Picking a legal move at random is easy with the legal move table, M$. If there are three possible moves then all we have to do is generate a number between one and three and extract the corresponding character from the string M$(P). The number of legal moves for the position P% is stored in M%(P). Line 6010 generates the random number between 1 and M%(P%) and line 6020 picks the correct character from M$(P%). This random move is then carried out by calling procedure do_move. The board is then printed and the whole cycle repeated, beginning with the check to see if the final position has been reached.

It might be imagined that such a simple problem could be solved by computer using random moves in a fairly short time. After all, the computer works so fast that it can afford to make many hundreds of wrong moves before finally making a right one. The truth is that even though a computer ▶

```
8000 DEFine PROCedure one_move
8010 C%=-5
8020 FOR I=1 TO M%(P%)
8030  K=M$(P%,I)
8040  IF K<>Q% THEN
8050   dist_chnge
8060   IF E%>C% THEN
8070    J%=K
8080    C%=E%
8090   END IF
8100  END IF
8110 END FOR I
8999 END DEFine one_move
9000 DEFine PROCedure dist_chnge
9010 E%=P%-B%(K)-INT((P%-1)/3)*3
9020 E%=ABS(E%+INT((B%(K)-1)/3)*3)
9030 E%=E%+ABS(INT((P%-1)/3)-INT((B%(K)-1)/3))
9040 F%=K-B%(K)-INT((K-1)/3)*3
9050 F%=ABS(F%+INT((B%(K)-1)/3)*3)
9060 F%=F%+ABS(INT((K-1)/3)-INT((B%(K)-1)/3))
9070 E%=F%-E%
9999 END DEFine dist_chnge
```

Figure 3. The two procedures which replace rnd_mve in the final version. Note that two lines replace line 90 from fig 2; '85 one_move' followed by '95 dist_chnge'.

can make many random moves it still takes a long time to find the answer. I have yet to see this simple method yield an answer even after running it one hundred thousand moves!

Common Sense

If you watch the random movements produced by the previous program you cannot help but despair as it fails to take even the most obvious move that would improve the situation. Even if it gets within one move of success it's just as likely to take the wrong option and start to disorder the pattern again! An observer quickly comes to the conclusion that there must be a better approach to the problem, even though no algorithm exists for its solution.

If we were to have some measure of how far away from the final arrangement the current arrangement was, then we could use the fairly obvious heuristic – select the move that takes the arrangement 'closer' to the desired arrangement. The only problem is what do we mean by 'closer'. Looking at the current position of any tile, you can describe how far it is away from its final position by counting how many horizontal and vertical moves it would take to move to that final position (ignoring the 'space'). For example the 6 in *figure 4* could be moved to its final position by one move down and two moves across – it's three moves from 'home'. Notice that it doesn't matter that you couldn't move the 6 along this route becase the 'space' is in the bottom left corner. We are only interested in using this as a rough measure of how far the 6 is from its final place. This measure is known as the 'city block distance' because it is the way distance is measured between two points if you have to walk via streets laid out at right angles. To summarise how far the entire board is from its final position we could work out how far every tile is from its final place and add up the total number of moves.

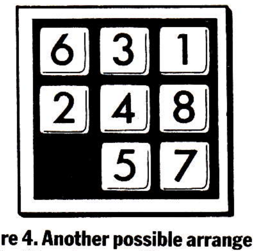


Figure 4. Another possible arrangement for the tile set-up positions.

So, we now have one possible meaning of the word 'closer' in our heuristic. We examine each of the possible moves and work out the distance between the final arrangement and the one that would result if we took the move. Obviously we should take the move that gets us closer to the final arrangement. In practice, working out the distance of the entire board takes rather too long and we can simplify things by noticing that the move to select is the one that produces the largest change in the distance toward the final arrangement. That is, we are not interested in the actual value of the distance, only in making it smaller. The practical result of this observation is that as we only move one tile at a time it is easy to find the change in the distance brought about. The new heuristic is:

- work out the current distance of the tile to be moved from its home position
- work out the new distance that the tile would be from its home position if it *was* moved
- work out the difference between the above, which gives the change in the overall distance if the tile was moved. Then choose the move that produces the largest change toward the final position.

Altering the tile program to adopt this heuristic involves writing two new procedures (one_move and dist_chnge) to replace the random move generator procedure rnd_mve. These are shown in *figure 3*. To run the new version of the program alter line 90 to call procedure one_move instead of rnd_mve.

Procedure one_move 'scans' through each possible move and calls procedure dist_chnge to work out the change in distance that would be produced if the move were made. The move with the largest change is picked by line 8060. The structure of the procedure dist_chnge may appear difficult but all that's being done is the calculation of city block distances for each position (lines 9010/9020 and 9040/9050) and then finding the differences (line 9070). *Table 1* summarises the program blocks.

If you run the new version of the program you might be lucky and see a solution in less than 100 moves. On the other hand you might not get a solution after thousands of moves – this is a heuristic not an algorithm and therefore cannot guarantee a solution. On average the program will solve one in three problems in under 100 moves, so if you are unlucky try again with a different board.

TABLE 1

PROCEDURE	DESCRIPTION
init	Defines arrays and legal move table
print_board	Prints board
set_shuffle	Shuffles the tile positions
find_space	Finds the current position of the 'space' and places it in P%
win_pos	Checks for winning position, S% = 0 indicates that final position has been reached
pause_1	General purpose delay procedure
rnd_mve	Select a move at random
do_move	Performs move of tile in M% (J%) to M%(P)
one_move	Finds 'best' move
dist_chnge	Calculates evaluation function

Building Success

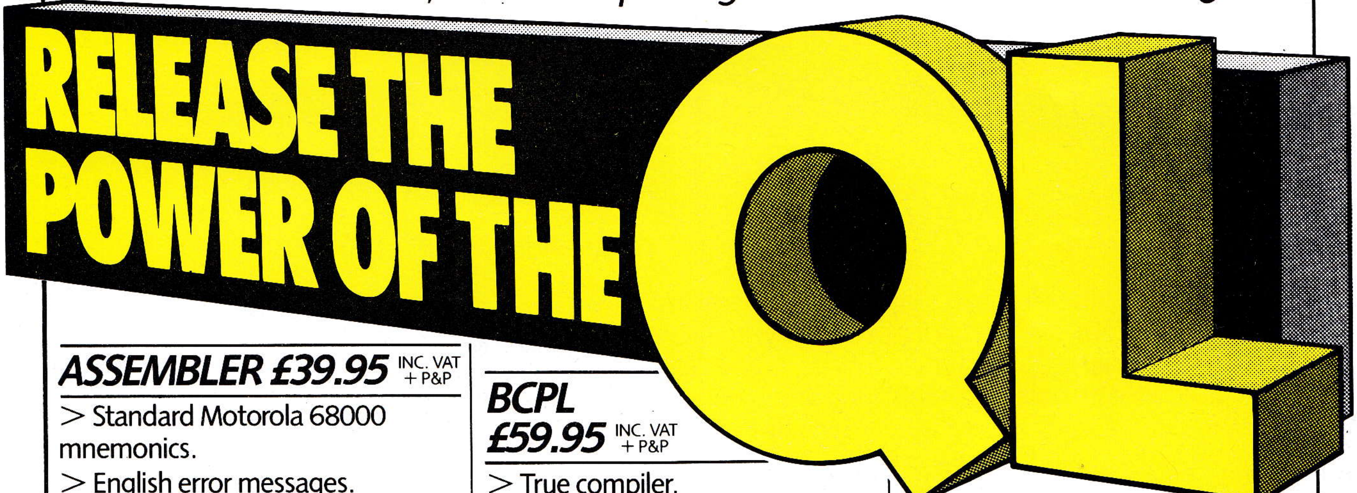
Changing from making random moves to applying the heuristic is very small in terms of programming but it does provide a dramatic improvement in performance. The example that the random move selection could not solve in over one hundred thousand moves was solved in 51 moves using the heuristic.

This does not mean there are no problems with the method. Some of these difficulties are instructive in themselves and deserve further study. For example, sometimes the program 'gets stuck' repeating the same set of moves over and over again. One reason for this is due to there being a contest for 'best move' when there's a tie. The first move in the legal move table is always taken. This could be changed, for example, to a random selection of tied moves (remember there is no guarantee that the heuristic will solve the problem, but it is better than random selection and considerably better than no solution at all).

The heuristic used to solve the tile game is very simple – 'examine each posible move and take the one that produces the maximum reduction in the distance between the current arrangement and the target arrangement'. This is intuitively sensible because each move tries to take us closer to the answer. However, if you watch a person playing the tile game you will notice that moves sometimes make the distance greater. What this suggests is that it is sometimes worth temporarily going away from the solution if this gains an advantage in later moves. The conclusion is that it isn't always enough to evaluate how good the very next move is. You have to evaluate a move in terms of what moves can follow it. In other words, you have to 'look ahead'. This is where the speed and power of computers really comes into their own. A problem can sometimes be solved by investigating the effects of making a long sequence of moves using a simple heuristic and this idea will form the subject of next month's article.

Write programs to unlock the multitasking power of your QL!

*Use Metacomco' s much acclaimed **Development Kits** to create the software for your QL: Use **Assembler** for its speed, and for access to the QL's many features; use **BCPL** for systems programming; writing games, utilities, and applications packages; use **LISP** for manipulating data structures, and for exploring the world of artificial intelligence.*



ASSEMBLER £39.95 INC. VAT + P&P

> Standard Motorola 68000 mnemonics.
> English error messages.
> Produces code which can be EXECed, and run as a concurrent job.
> External references allow linkage to high level languages.
> Integral linker allows assembler programs to be linked together.
> Macro expansions.
> Position independent, absolute or relocatable code can be produced.
> Conditional assembly.
> Large range of directives.
> Fully formatted listings.
> 30 character variable names and 32-bit values.

ENHANCED!
NOW INCLUDES ASSEMBLER LINKER*
AND NEW LOW PRICE £39.95

BCPL £59.95 INC. VAT + P&P

> True compiler.
> Multiple BCPL programs can run concurrently.
> Full runtime library includes interfaces to QDOS graphics, file operations, window handling, etc.
> Exception handling provides diagnostics for debugging.
> Link loader allows linkage of separately compiled segments.
> Assembler and BCPL modules can be linked together.
> 32-bit variables use the full QL address space.

Each language is supplied as a complete Development Kit which includes the software (on a Microdrive Cartridge), Metacomco's popular full screen editor, and user manual.

LISP £59.95 INC. VAT + P&P

> Full support of QL features including window graphics, and screen handling.
> Compatible with Acornsoft Lisp for the BBC micro.
> Interpreter.
> Turtle graphics.
> Structure editor allows alteration of data structures.
> Prettyprinter displays programs in structured format.
> Tracer to aid in debugging.
> Garbage collector automatically recovers spare memory space.
> 28-bit integers and 250-character names.

The Metacomco Development Kits are available from HMV, Menzies and many other leading retailers, or direct from Metacomco.

METACOMCO

26 Portland Square, Bristol BS2 8RZ.
Tel: Bristol (0272) 428781

*Existing registered users can obtain an upgrade. Contact Metacomco for details.

QL & QDOS are the trade names of Sinclair Research Ltd.

NAME ____________________

ADDRESS ____________________

____

POSTCODE ____________________

TEL. NO. ____________________

SIGNATURE

____

Phone today, or post this coupon to: Metacomco, 26 Portland Square, Bristol BS2 8RZ.

Please send me:

The Assembler Development Kit at £39.95 ________

The BCPL Development Kit at £59.95 ________

The LISP Development Kit at £59.95 ________

I enclose a cheque for £ ________ or debit my ACCESS/VISA Account No.

□□□□□□□□□□□□□□□□

Card expiry date: ________

Price includes VAT, postage and packing UK mainland only. Delivery allow up to 28 days. QLU6 silicon express

INSIDER DISK INTERFA

Our new QL disk interface is a breakthrough in modern technological design. It is the first in a range of INSIDER BOARDS manufactured by Silicon Express, utilising the tremendous expansion potential of Sinclair's professional, personal computer. The software has been designed for us by the author of QDOS to ensure total compatibility with all existing QL file commands, machines and Psion software revision packages.

Unlike other disk interfaces already on the market, our INSIDER BOARD fits snugly into the case of the QL, connecting to the main expansion port, so there's no problem with protruding PCB's which, if knocked, could lose you irreplaceable data and which can add over 2" to the length of your microcomputer.

This double density system is compatible with all 3", 3½' and 5¼" independently-powered disk drives (although we recommend our own for added data security). It can support two drives in any combination of 40 or 80 track, single or double-sided, thereby offering you a storage capacity of 1.4 megabytes (formatted). Typically, it allows you to boot 'Quill' in under 10 seconds. The system is expandable to allow you to further upgrade your QL at a later date.

You can purchase the INSIDER BOARD on its own for £99 + VAT or as part of a disk package with prices starting from £217 + VAT. In addition, we are offering double-sided double density 96TPI coloured diskettes – the ideal storage media for your valuable data.

For immediate despatch of goods you can order now by phoning us on Leicester (0533) 374917 quoting your Access card number.

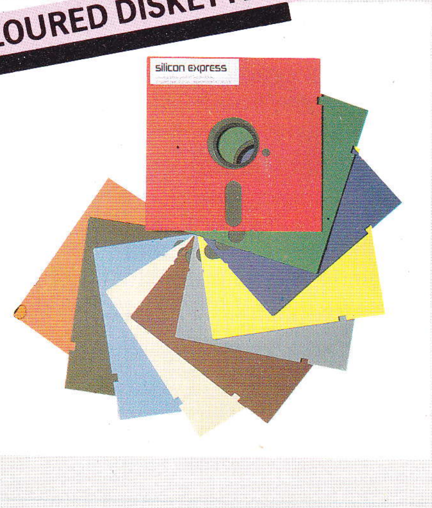
Or fill in and post the coupon — no stamp required — and we will despatch your goods within 14 days.

The interface carries a full two year warranty and is supplied complete with a comprehensive user manual.

Sinclair, QL and QDOS are registered trade marks of Sinclair Research Ltd.

QL DISK DRIVE

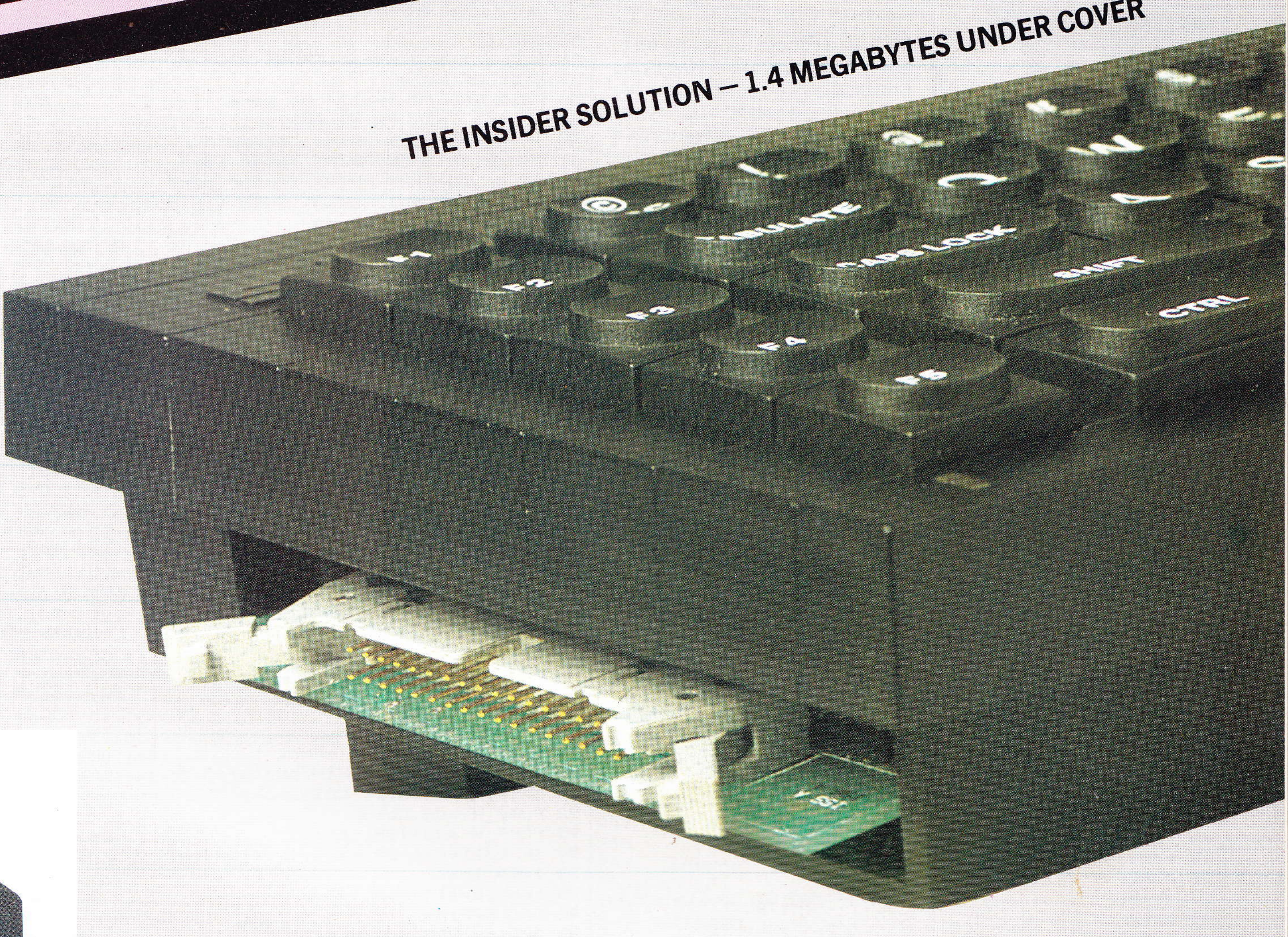
COLOURED DISKETTES

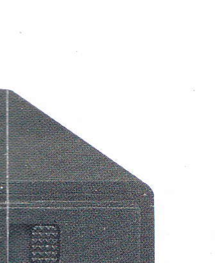


INS

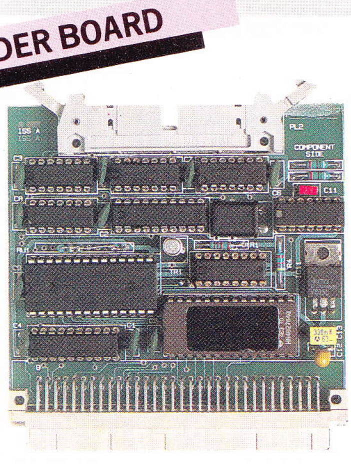
ACE CARD....

THE INSIDER SOLUTION – 1.4 MEGABYTES UNDER COVER





NSIDER BOARD



To order please phone Leicester (0533) 374917 or complete and post this coupon.

ORDER FORM No stamp required if posted in the UK

Post to: Silicon Express Ltd., FREEPOST, Rothley, Leicester, LE7 7QZ.

Please allow 14 days for delivery of postal orders; telephone orders, immediate despatch.

_____ (Qty) Insider Boards at £113.85 ....£

_____ (Qty) Insider Board with single, 80 track, double sided 5¼" drive at £249.00 ....£

_____ (Qty) Insider Board with dual, 80 track, double sided 5¼" drive at £399.00 ....£

_____ (Qty) Box of 10 DSDD 96TPI coloured diskettes at £35.00 ....£

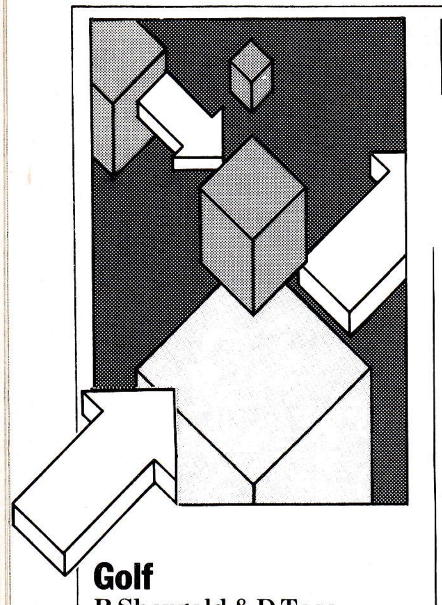
_____ Further details on the INSIDER BOARD

I enclose cheque/postal order for £ made payable to Silicon Express Ltd.

Name

Address

Prices include VAT, post and packing. QU5



THE PROGS

Sprite design is the theme of this month's section – programs of all descriptions are considered. Send them to 'The Progs', QL User, Priory Court, 30-32 Farringdon Lane, EC1R 3AU.

Golf

R Shergold & D Tose

What's extraordinary about this golf simulation is that it packs in so much detail. Moving from fairway to green, you can choose to play on up to 50 different courses of varying difficulty. The usual hazards abound – lakes, rivers, bunkers and trees. The course record is set with opportunities to go for birdies, eagles and even albatrosses(?). Your scorecard may be saved and congratulations are in order if you beat the likes of Faldo, Watson, Ballesteros or Kite. Finally, to play the game you use the space bar to toggle the power behind your stroke and enter clock directions. For example, an entry of "3.0" would send your ball careering from left to right across the screen along the horizontal.

```
1 REMark **** GOLF - QL US
2 REMark
3 REMark R.Shergold and D.Tose
4 BORDER 0,0:SCALE 100,0,0:PRINT "Please Wait"
5 holes = 9:initialise:CLS:max = 100
6  CLS#0:BORDER#0,2,4:course=0:CLS#0
7 INPUT#0,\"Load Your Own High Scores ? (y/n) ";an
$
8 IF an$="Y" OR an$="y" THEN dataload
9 CLS#0:PRINT#0;\"SPACE=PLAY : ANY KEY=TOPSCORES"
10 pointer$=INKEY$(-1)
11 IF pointer$<>" " THEN top
12 pointer$=" ":CLS#0
13 INPUT#0;"Number of course? (1-100)  ";course
14 IF course>100 OR course<0 THEN GO TO 13
15 RANDOMISE course
16 DIM shoots(holes):DIM pars(holes)
17 PAPER#0,0:CLS:sc1=0
18 FOR hole = 1 TO holes
19 shots=0:finished=0
20 PAPER 7,4,1:CLS:CLS#0
21 trees 75:fairway:trees 45:IF RND(0 TO 1)=1 THEN
22 water
23 ELSE
24 lake
25 END IF
26 green
27 length=800-20*c+RND(0 TO 20)
28 IF length<=270 THEN  par =3
29 IF length>270 AND length<=395 THEN par =4
30 IF length >395 THEN  par=5
31 CSIZE 0,0: INK 0:PRINT "  Hole ";hole;
32 PRINT "       Length ";length;
33 PRINT "       Par ";par
34 AT 19,0:PRINT "score ";sc1;"        Course ";cou
rse
35 REPeat club
36 pow:PAUSE 50:ang:shots=shots+1:shoot:test
37 IF finished THEN EXIT club
38 END REPeat club
39 PAPER 1:INK 7:CLS#0
40 PRINT#0,"      It's in, well played":PAUSE 40:CL
S
41 shoots(hole)=shots:pars(hole)=par
42 IF shots=1 THEN hole_in_one
43 IF shots<=par THEN
44 CLS#0:r=par-shots:SELect ON r
45 ON r=0:PRINT#0, "A Par, Not Bad"
46 ON r=1:PRINT#0, "Well played, A birdie"
47 ON r=2:PRINT#0, "Fantastic shot, An Eagle"
48 ON r=3:PRINT#0, "An Albatross..incredible!"
49 END SELect
50 shoots(hole)=shots:pars(hole)=par
51 scorecard
52 ELSE :CLS#0
53 PRINT#0, shots-par!"over par on this hole."
54 shoots(hole)=shots:pars(hole)=par:scorecard
55 END IF
56 sc1=sc1+shots-par:PAUSE 200
57 END FOR hole
58 PRINT#0, "Overall you were ";
59 IF sc1 THEN PRINT ABS(sc1);" ";
60 IF sc1=0 THEN PRINT "on par"
61 IF sc1>0 THEN PRINT "over par"
62 IF sc1<0 THEN PRINT "under par"
63 IF sc1<score(course) THEN top:datasave
64 CLS#0:CLS:scorecard
65 CLS#0:PRINT#0,"Hit Space Bar To Play Again"
66 IF INKEY$(-1)=" " THEN GO TO 7
67 GO TO 65
68   DEFine PROCedure pow
69 CLS#0:INK#0,7
70  PRINT#0,"                  0        50       100"
71 AT#0, 0,0:PRINT#0, "Space Bar "\"Selects"
72 FOR p = 0 TO 200
73 IF CODE (INKEY$)=32 THEN GO TO 76
74 BLOCK#0,2,5,p+200,20,2:NEXT p
75 IF INKEY$(30)<>" " THEN GO TO 69
76 power=INT(p/2)
77 IF power=0 THEN GO TO 69
78 END DEFine
79 DEFine PROCedure ang
80 CLS #0
81 PRINT #0;"angle (0-12)"\\"e.g 12 = >"
82 AT#0,0,14:INPUT#0,angle:angle=angle*PI/6
83 IF power >max THEN
84 CLS#0:PRINT#0;"You Have Fluffed It!":power=powe
r/4
85 angle=angle+RND(-20 TO 20)/100:PAUSE 30
86 max =100
87 END IF
88 END DEFine
89 DEFine PROCedure shoot
90 ballx=INT(ballx+SIN(angle)*525*power/length)
91 bally=INT(bally-COS(angle)*525*power/length)
92 POKE_W add,color
93 POKE_W add+128,color1
94 IF bally<26 OR bally>204 OR ballx<16 OR ballx>2
38 THEN lost
95 look:ball
96 END DEFine
97 DEFine PROCedure ball
98 add=INT(131072+128*bally+INT(ballx/4)*2)
99 color=PEEK_W (add)
100 color1=PEEK_W (add+128)
101 bit=(ballx MOD 4)*2:bitt=(255-2^(7-bit)*1.875)
102 IF bitt-INT(bitt) THEN bitt=bitt*1.6
103 newcol=(bitt && PEEK (add))*256+(bitt && PEEK
(add+1))
104 POKE_W add,newcol
105 newcol1=(bitt && PEEK (add+128))*256+(bitt &&
PEEK (add+129))
106 POKE_W add+128,newcol1
107 FOR n=0 TO 4
108 POKE_W add,color:POKE_W add+128,color1
109 PAUSE 10
110 POKE_W add,newcol:POKE_W add+128,newcol1
111 PAUSE 9:END FOR n
112 END DEFine
113 POKE_W add,color:POKE_W add+128,color1
114 PAUSE 10
115 POKE_W add,newcol:POKE_W add+128,newcol1
116 PAUSE 9:END FOR n
117 END DEFine
118 DEFine FuNction getcol(x,y)
119 LOCal add
120 add=INT(131072+128*y+INT(x/4)*2)
121 bit=(x MOD 4)*2:bitt=(2^(7-bit))
122 grn=4*((bitt && PEEK (add))<>0)
123 bitt=(2^(7-bit)*1.5):rb=(bitt && PEEK (add+1))
/bitt*3
124 rb=(bitt && PEEK (add+1))/bitt*3:RETurn grn+rb
125 END DEFine
126 DEFine PROCedure fairway
127 INK 4:a=50:b=25:blockx=5:blocky=b+90
```

```
128 FILL 0
129 FOR f = 0 TO 170 STEP .7
130 b=INT(b-1+RND(0 TO 2))
131 IF INT(f)=150 THEN LET c=b:e=a
132 IF f> 70 AND RND (0 TO 100)<2 THEN
133 INK 6
134 FILL 1:ELLIPSE f-10,a-RND (0 TO b),4,.7,RND(0
TO 7)
135 INK 4:FILL 0
136 END IF
137 IF b<15 THEN LET b=15
138 IF b>35 THEN LET b=35
139 IF a-b<10 THEN LET a=10+b
140 IF a>90 THEN a=90
141 LINE f,a TO f,a-b
142 a=INT(a-2+RND(0 TO 4))
143 END FOR f
144 END DEFine
145 DEFine PROCedure green
146 FOR g = 1 TO 5:FILL 1:INK 6,4,3
147 ELLIPSE 150,e-c/2,(c-g)/2,.7-RND(0 TO .1),RND(
0 TO 10)
148 IF g/2 = INT(g/2) THEN END FOR g
149 INK 6:FILL 1
150 x=RND(0 TO c)-c/2:y=SQRT(c^2/4-x^2)
151 aa=RND(0 TO 1):IF aa<.5 THEN LET y = -y:
152 ELLIPSE x+150,y+e-c/2,c/5,.6-RND(0 TO .2),RND(
0 TO 10)
153 END FOR g
154 zx=RND(-c/4 TO c/4):xz=RND(-c/4 TO c/4):FILL 0
155 INK 0:CIRCLE 150+zx,e-c/2+xz,.6:INK 7
156 LINE 150+zx,e-c/2+xz TO 150+zx,e-c/2+c/5+xz
157 INK 2:LINE 150+zx,e-c/2+c/5+xz TO 152+zx,e-c/2
+c/6+xz
158 LINE 152+zx,e-c/2+c/6+xz TO 150+zx,e-c/2+c/6+x
z
159 holex=150+zx:holey=e-c/2+xz
160 holex=INT((holex/166*448+33)/2)
161 holey=INT((100-holey)*2+16.5)
162 BLOCK c/2.5,c/2.5,blockx-3,blocky,7
163 ballx=19:bally=129+INT(c/4)
164 ball:END DEFine
165 DEFine PROCedure water
166 y=0:x=RND (50 TO 350)
167 BLOCK 7,4,x,y,5
168 y=y+4:x=x-3+RND (0 TO 7)
169 IF y=200 THEN GO TO 171
170 GO TO 167
171 END DEFine
172 DEFine PROCedure lake
173 IF RND(0 TO 2)=1 THEN END DEFine
174 FILL 1:INK 5
175 x=RND(30 TO 175):y=RND(30 TO 70)
176 FOR o=-2 TO 2:ji=y+RND(-5 TO 5)
177 ELLIPSE x+(4*o),ji,20,.65,RND(1 TO 7)
178 NEXT o
179 FILL 0
180 END DEFine
181 DEFine PROCedure trees(ee)
182 INK 4,3,3
183 FOR f = 1 TO ee
184 FILL 1:ok=RND(15 TO 75)
185 CIRCLE RND(1 TO 175),ok,1+RND(0 TO 10)/7
186 FILL 0
187 END FOR f
188 END DEFine
189 DEFine PROCedure look
190 c1=getcol (ballx,bally)
191 c2=getcol (ballx+1,bally)
192 c3=getcol (ballx,bally+1)
193 c4=getcol(ballx,bally+1)
194 END DEFine
195 DEFine PROCedure test
196 hazard=0:bm=ABS(ballx-holex)
197 IF bm<=1 AND ABS(bally-holey)<=1 THEN
198  POKE_W add,color:POKE_W add+128,color1:finish
ed=1:END DEFine
199 END IF
200 IF c1=6  AND c2=6 OR c1=6 AND c3=6 THEN bunker
201 IF c1=5 THEN wet
202 IF c1=7 AND c3=4 OR c1=4 AND c3=7 THEN rough
203 IF c1=4 AND c2=4 OR c1=4 AND c3=4 THEN END DEF
ine
204 IF c1=3 OR  c2=3 THEN branches
```

```
205 IF c1=7 AND c2=7 THEN END DEFine
206 put
207 hazard=0:PAUSE 100:CLS#0
208 END DEFine
209 DEFine PROCedure bunker
210 IF hazard THEN END DEFine
211 hazard=1
212 CLS#0:PRINT#0\"   BUNKERED!"
213 max=25:END DEFine
214 DEFine PROCedure wet
215 IF hazard THEN END DEFine
216 hazard =1:CLS#0
217 PRINT#0;"WATER !!!. Back to tee"
218 ballx=19:bally=129+INT(c/4)
219 POKE_W add,color:POKE_W add+128,color1
220 ball
221 END DEFine
222 DEFine PROCedure rough
223 IF hazard THEN END DEFine
224 hazard =1
225 CLS#0:PRINT#0;"You Are In The Rough"
226 max =75:PAUSE 50
227 END DEFine
228 DEFine PROCedure put
229 REPeat putting
230 dist=SQRT((holey-bally)^2+(holex-ballx)^2)
231 dist=dist/(224/length)
232 dist=INT(dist*100)/100
233 IF dist>50 THEN END DEFine
234 IF hazard THEN END DEFine
235  hazard=1:CLS#0
236 PRINT#0;"You Are ";dist;" yards  from the hole
"
237 PAUSE 100:BLOCK#0,440,4,0,32,4
238 OVER#0,-1
239 angle=ATAN(ABS(holex-ballx)/ABS(bally-holey+1E
-5))
240 IF holey>bally THEN
241   IF holex<=ballx THEN
242     angle=angle+PI
243   ELSE
244     angle=PI-angle
245   END IF
246 ELSE
247   IF holex<=ballx THEN
248     angle=2*PI-angle
249   END IF
250 END IF
251 putx=31:puty=24
252 CURSOR#0, putx,puty:PRINT#0, "o"
253 FOR ang1=0 TO 60 STEP 2
254 FOR plot=0 TO 1
255 LINE#0,60,72 TO 62-62*SIN(RAD(ang1)),72-62*COS
(RAD(ang1))
256 END FOR plot
257 IF INKEY$=" "THEN GO TO 260
258 END FOR ang1
259 IF INKEY$(10)<>" " THEN GO TO 253
260 FOR a=ang1 TO 0 STEP -2
261 FOR plot=0 TO 1
262 LINE#0,60,72 TO 62-62*SIN(RAD(a)),72-62*COS(RA
D(a))
263 END FOR plot
264 END FOR a:CURSOR#0,putx,puty:PRINT#0;"o"
265 power=ang1/7:shoot:test:shots=shots+1
266 PAUSE 1+49*NOT finished
267 IF finished THEN EXIT putting
268 END REPeat putting
269 OVER#0,0
270 END DEFine
271 DEFine PROCedure branches
272 IF hazard THEN END DEFine
273 CLS#0:PRINT #0;"You Have Hit A Tree!"
274 ballx=ballx-2: bally=bally+RND(-1 TO 1)
275 POKE_W add+128,color1:look:ball:test
276 hazard=1
277 END DEFine
278 DEFine PROCedure lost
279 CLS #0
280 PRINT #0;"Ball Out Of Bounds"\"penalty shot"
281 IF ballx<16 THEN ballx=16
282 IF ballx>238 THEN ballx=238
283 IF bally>204 THEN bally=204
284 IF bally<26 THEN bally=26
```

```
285 shots=shots+1:PAUSE 50
286 END DEFine
287 POKE_W add,color:POKE_W add+1,color1
288 holed=1:END DEFine
289 DEFine PROCedure hole_in_one
290 PAPER 1:INK 7:CLS
291 INK 7:FOR f= 1 TO 130 STEP 9
292 CURSOR f,50:PRINT " Hole I":NEXT f
293 FOR f= 320 TO 212 STEP -9:CURSOR f,50:PRINT "n
 One ":NEXT f
294 PRINT \\"              Superb Shot!!!":PAUSE 200:
CLS
295 PAUSE 200:CLS
296 END DEFine
297 PAPER 0:CLS:CSIZE 3,1:INK 7:CLS#0
298 CLS#0:PAPER 1:INK 7:CSIZE 0,0:CLS
299 DEFine PROCedure top
300 PAUSE 100:CLS#0:CLS
301 IF pointer$<>" " THEN GO TO 305
302 PRINT "Well done-a course record"
303 INPUT#0,"Your name? ";name$(course)
304 score(course)=sc1:PAPER 1:INK 7:CLS
305 start=1:finish=50:MODE 512
306 CLS:FOR f = start TO finish
307 IF f=course THEN INK 2:PAPER 6
308 PRINT f;"  ";name$(f,1 TO 10);"  ";score(f),
309 INK 7:PAPER 0:NEXT f
310 IF start = 51 THEN
311 CLS#0:CSIZE#0,1,0:PRINT #0,\"
 Hit Any key"
312 PAUSE 10000:MODE 256:CSIZE#0,0,0:END DEFine
313 ELSE
314 CLS#0:CSIZE#0,1,0:PRINT#0,\"               Hit
any Key to see next 50"
315 PAUSE 10000:start=51:finish=100:CSIZE#0,0,0:GO
 TO 306:END DEFine
316 DEFine PROCedure scorecard
317 PRINT\
318 PRINT " Hole","Par","Score"
319 PRINT:tot=0:totpar=0
320 FOR f= 1 TO holes
321 dif=shoots(f)-pars(f)
322 SELect ON dif=1 TO 20
323 ON dif=-1:d$="Birdie"
324 ON dif=-2:d$="Eagle"
325 ON dif=-3:d$="Albetros"
326 ON dif=0:d$="Par"
327 ON dif=1 TO 20:d$="Bogey "& dif
328 END SELect
329 IF shoots(f)=0 THEN d$="------"
330 PRINT "  ";f,pars(f)," ";shoots(f);"         ";
331 INK 7:PRINT d$:INK 7
332 tot=tot+shoots(f):totpar=totpar+pars(f)
333 NEXT f
334 PRINT \,totpar," ";tot,"  ";:PRINT tot-totpar
335 PRINT\\"Course Record ";score(course);" by ";
336 PRINT name$(course,1 TO 10)
337 END DEFine
338 DEFine PROCedure datasave
339 DELETE mdv1_golf_data
340 OPEN_NEW #9,mdv1_golf_data
341 FOR n=1 TO 100:PRINT#9,score(n):NEXT n
342 FOR n=1 TO 100:PRINT#9,name$(n, 1 TO 10)
343 NEXT n
344 CLOSE#9
345 END DEFine
346 DEFine PROCedure dataload
347 OPEN_IN #9,mdv1_golf_data
348 FOR n= 1 TO 100:INPUT#9,score(n):NEXT n
349 FOR n= 1 TO 100:INPUT#9,name$(n,1 TO 10)
350 NEXT n
351 CLOSE#9
352 END DEFine
353 DEFine PROCedure initialise
354 DIM cse(100):DIM score(100):DIM name$(100,16)
355 RANDOMISE 20000:RESTORE 6
356 FOR f= 1 TO 100
357 LET score(f)=RND(-9 TO -4):READ player$
358 IF (f/5)=INT(f/5) THEN RESTORE 360
359 name$(f,1 TO 10)=player$
360 DATA "N.Faldo   ","T.Watson   ","T.Kite    ","B
alesteros","G.Norman"
361 NEXT f
362 END DEFine
```

3Dscapes

Rob Miles

There are two major ways of 'looking' at a three dimensional surface, one is called 'isometric' and the other is called 'perspective'. This program allows you to explore these two techniques, which produce some very attractive surfaces. The height of each part of the surface is given at line 730 in the function height. By plugging your own equations in at this line you can produce wildly different results – try to ensure that the value returned does not go outside the range +2 to −2 for best results.

a) 730 a = x-halfx: b = y-halfy:z = EXP (COS(SQRT (a★a + b★b)/3))★1.75

b) 730 a = x-halfx: b = y-halfy:d = SQRT (a★a + b★b):Z = COS (d/7)★ABS (6-d)

c) 730 d = SQRT (x★y):z = INT (SIN(d/20★PI)★20) − 10

d) 730 z = x MOD 5

e) 730 a = x-halfx:b = y-halfy:d = a★a + b★b:z = SIN (SQRT (d))

f) 740 a = x-halfx:b = y-halfy:z = SQRT (a★a + b★b) − 15

```
100 RESTORE
110 REMark 3D Plotter - QL User May 1985
120 REMark (c) 1985 Rob Miles
130 xlim=20:ylim=20:halfx=xlim/2:halfy=ylim/2
140 ximult=140/(xlim+ylim):yimult=100/(xlim+ylim)
150 xpmult=24:ypmult=50
160 DIM vx(xlim,ylim),vy(xlim,ylim)
170 REPeat disploop
180    CLS
190    setup
200    plot
210    picsave
220 END REPeat disploop
230 REMark
240 DEFine PROCedure plot
250 IF perspective<2 THEN
260    REMark Perspective Projection
270    CLS
280    setrow xlim:off=0
290    FOR x=(xlim-1) TO halfx STEP -1
300       cell x,ylim
310       FOR y=(ylim-1) TO 1 STEP -1
320          cell x,y
330          lay x,y,1,1
340       NEXT y
350    NEXT x
360    setrow 0:off=1
370    FOR x=1 TO halfx
380       cell x,ylim
390       FOR y=(ylim-1) TO 1 STEP -1
400          cell x,y
410          lay x,y,-1,1
420       NEXT y
430    NEXT x
440 ELSE
450    REMark Isometric Projection
460    CLS
470    setrow 0:off=0
480    FOR x=1 TO xlim
490       cell x,ylim
500       FOR y=ylim-1 TO 0 STEP -1
510          cell x,y
520          lay x,y,-1,1
530       NEXT y
540    NEXT x
550    END IF
560 END DEFine plot
570 REMark
580 DEFine PROCedure lay (x,y,dx,dy)
590 INK 3+((x+y+off) MOD 2)
600 POINT vx(x,y),vy(x,y):FILL 1
610 LINE TO vx(x,y+dy),vy(x,y+dy)
620 LINE TO vx(x+dx,y+dy),vy(x+dx,y+dy)
630 LINE TO vx(x+dx,y),vy(x+dx,y)
640 LINE TO vx(x,y),vy(x,y)
650 FILL 0:INK 7
660 LINE vx(x,y),vy(x,y) TO vx(x,y+dy),vy(x,y+dy)
670 LINE TO vx(x,y+dy),vy(x,y+dy)
680 LINE TO vx(x+dx,y+dy),vy(x+dx,y+dy)
690 LINE TO vx(x+dx,y),vy(x+dx,y)TO vx(x,y),vy(x,y
)
700 END DEFine lay
710 DEFine FuNction height (x,y)
```

```
720 REMark Height equation goes here
730 a=x-halfx:b=y-halfy:d=SQRT(a*a+b*b):z=COS(d/7)
*ABS(6-d)
740 IF invert<2 THEN z=z*-1
750 RETurn z
760 END DEFine height
770 DEFine PROCedure cell(x,y)
780 z=height(x,y)
790 IF perspective<2 THEN
800     s=x-halfx:z=z-5:d=SQRT(SQRT(z*z+s*s)+y*y)
810     vx(x,y)=(s/d)*xpmult+72
820     vy(x,y)=(z/d)*ypmult+110
830 ELSE
840     vx(x,y)=(x+y)*ximult+3
850     vy(x,y)=(z-x/2+y/2)*yimult+50
860 END IF
870 END DEFine cell
880 DEFine PROCedure setrow(x)
890 FOR y=0 TO ylim
900   cell x,y
910 NEXT y
920 END DEFine setrow
930 DEFine PROCedure setup
940 MODE 4
950 WINDOW 512,256,0,0
960 BORDER 2,4:PAPER 0:INK 7:CSIZE 1,0:CLS
970 UNDER 1:AT 2,20:PRINT "QL Surface Projection"
980 UNDER 0:AT 5,8:PRINT "Perspective (P) ";
990 INPUT "or Isometric (I) projection ?";per$
1000 perspective=per$ INSTR "PpIi"
1010 IF perspective=0 THEN GO TO 980
1020 AT 10,15:INPUT "Invert (Y) or (N) ?";inv$
1030 invert=inv$ INSTR "YyNn"
1040 IF invert=0 THEN GO TO 1020
1050 AT 12,15:INPUT "Filename for save ";f$
1060 AT 13,15:PRINT "Press S at tone to save"
1070 AT 15,15:PRINT "Press any key to start plot"
1080 d$=INKEY$(-1)
1090 END DEFine setup
1100 DEFine PROCedure picsave
1110 BEEP 15000,4,12,40,1
1120 d$=INKEY$(-1)
1130 IF d$ INSTR "Ss" THEN SBYTES f$,131072,32768
1140 END DEFine picsave
```

Sound Experimentor

Rob Miles

Unlike most home computers the QL has a very limited noise making capability. What there is can only be dragged out by using the enigmatic BEEP command. This has 8 arguments, all of which are obscure. The only way to produce something vaguely melodic is by trial and error. The following quickie allows you to do exactly this.

```
100 REMark **** QL User -Beep experiment
110 REMark ****  Rob Miles 1985
120 duration=5000:pitch=100:pitch_2=0:grad_x=0:gra
d_y=0:wrap=0:fuzzy=0:random=0
130 REPeat sounder
140 CLS
150 set "Duration (-32768..32767) ?",duration,2
160 set "Pitch (0..255) ?",pitch,4
170 set "Pitch_2 (0..255) ?",pitch_2,6
180 set "Grad_x (-32768..15) ?",grad_x,8
190 set "Grad_y (-8..7) ?",grad_y,10
200 set "Wrap (0..32767) ?",wrap,12
210 set "Fuzzy (0..15) ?",fuzzy,14
220 set "Random (0..15) ?",random,16
230 AT 18,0:PRINT "Beep ";duration:",";pitch;",";p
itch_2:",";grad_x;",";grad_y;",";wrap;",";fuzzy;",
";random
240 BEEP duration,pitch,pitch_2,grad_x,grad_y,wrap
,fuzzy,random
250 dum$=INKEY$(-1)
260 END REPeat sounder
270 DEFine PROCedure set (name$,var,position)
280 LOCal buf$
290 AT position,0:PRINT name$;" ";var
300 AT position,0:PRINT name$;" ";
310 INPUT buf$
320 IF buf$<>"" THEN var=buf$
330 AT position,0:PRINT name$;" ";var:"      "
340 END DEFine set
```

Easel Print

D Duncan

The following 7 liner shows exactly how to use the graphics dump provided with EASEL. All you need is an EPSON compatible printer and the EASEL cartridge in mdv2.

```
120 MODE 8:PAPER 2:CLS
130 FOR n=1 TO 150
140 INK 7:FILL 1:CIRCLE n,50,20,.5,-n/2
150 INK 2:FILL 0:CIRCLE n,50,20,.5,-n/2
160 NEXT n
170 REMark Dump to Epson printer
180 a=RESPR(1024):LBYTES mdv2_gprint_prt,a
190 CALL a:OPEN #3,ser1:LIST #3:CLOSE #3
```

Programmable Function Keys

Richard Cross

Those acquainted with the BBC micro will realise what a bitter disappointment the QL's dumb function keys are. However, this ingenious program remedies all shortcomings. Simply type in the SuperBasic listing below and RUN it. This creates a machine code routine which will automatically be saved on mdv1 under the name of "key_bytes". Then, to use the routine proper, you need to load it into the QL's memory and call it as follows:

a = resp (1000)
lbytes mdv1_key_bytes, a
call a

The routine scans the functions keys (on a 50 Hz interrupt) and puts in a user-defined string by using the new command:

FKEY keynumber, string

keynumbers range from 1-10 covering F1-F5 with or without [SHIFT]. A string can contain a maximum of 48 characters and returns 'Buffer Full' if this limit is exceeded. A line feed may be included by using the following method:

FKEY 1, "mode 4:list"&chr$ (10)

This means whenever F1 is pressed you will be switched into hi-res mode and the current program will list automatically.

```
100 REMark ****  Function key definer
110 REMark **** QL USER MAY 1985
120 REMark **** Richard Cross
150 RESTORE 210
160 a=RESPR(1000)
170 CLS:PRINT "storing machine code"
180 FOR b=a TO a+330:READ x:POKE b,x
190 DELETE mdv1_key_bytes:SBYTES mdv1_key_bytes,a,
330
200 STOP
210 DATA 52,120,1,16,67,250,0,38,78,146,48
220 DATA 60,0,249,67,250,1,56,50,252,0,0
230 DATA 81,200,255,250,67,250,0,138,65,250,1
240 DATA 32,33,73,0,4,112,28,78,65,78,117
250 DATA 0,1,0,14,4,70,75,69,89,0,0
260 DATA 0,0,0,0,0,81,141,52,120,1,18
270 DATA 78,146,102,82,12,67,0,1,102,78,63
280 DATA 54,152,0,38,77,80,141,52,120,1,22
290 DATA 78,146,102,60,12,67,0,1,102,56,50
300 DATA 31,12,65,0,10,110,52,83,65,69,250
310 DATA 0,218,194,252,0,50,213,193,71,234,0
320 DATA 2,73,246,152,2,54,54,152,0,103,14
330 DATA 12,67,0,48,110,24,83,67,22,220,81
340 DATA 203,255,252,52,182,152,0,112,0,78,117
350 DATA 112,241,78,117,112,252,78,117,112,251,78
360 DATA 117,112,17,71,250,0,134,78,65,2,1
370 DATA 0,59,103,114,74,57,0,3,215,46,102
380 DATA 112,80,249,0,3,215,46,116,0,8,1
390 DATA 0,0,103,2,116,3,8,1,0,3,103
400 DATA 2,116,0,8,1,0,3,103,2,116,1
410 DATA 8,1,0,4,103,2,116,2,8,1,0
420 DATA 5,103,2,116,4,47,2,112,17,71,250
430 DATA 0,70,78,65,36,31,12,1,0,1,110
440 DATA 46,102,2,90,66,67,250,0,68,196,252
450 DATA 0,50,211,194,54,17,84,137,36,121,0
460 DATA 2,128,76,74,67,103,18,18,25,56,120
470 DATA 0,224,78,148,83,67,96,240,81,249,0
480 DATA 3,215,46,78,117,0,0,9,1,0,0
490 DATA 0,0,0,2,9,1,0,0,0,0,7
500 DATA 2,0,0,32,97,100,100,114,101,0,0
510 DATA 105
```

QL BANK ACCOUNT

Fed up with infrequent Bank statements? Avoid Bank charges and keep up to dte with QL BANK ACCOUNT. Check your mortgage HP, standing orders! Budget your bills.

★ Automatic posting of Standing Orders & other regular payments or credits.

★ Delete, amend and even backdate entries. Scroll through accounts.

★ 6 different entry search routines!

★ Post items to account headings chosen by YOU to suit your circumstances. Print out individual accounts & summary. eg all payments to Building Society.

★ Output of Bank statement & accounts to most makes of printer. Makes automatic use of memory expansions. Easily converted to disc.

Cartridge & Manual £19.95 + 50p p&p.

Cheques & POs to:

cenprime software
Cenprime Ltd., Dept. 1
10 Castle Street, RUGBY CV21 2TP
☎ **Coventry (0203) 686162**

QUALITY QL PERIPHERALS BARGAIN PRICES

Complete with cable ready to plug in. No untidy external interface. The printer which is DEDICATED to the QL; TOTALLY QL compatible.

SUPERB NEW SHINWA CPA80-Q

Fantastic 13 × 9 Dot Matrix	(note Epson is 9 × 9!)
100 CHARACTERS PER SECOND	Bi-Directional printing
True Descenders on y, p, j, etc.	Underlining on command
Backspacing	Up to 96 columns
Friction AND Tractor feed built-in	Normal, Double size & Italic print styles

Many, many other advanced features.
Delivery, VAT and Insurance included in our price.
NO HIDDEN EXTRA COSTS
ONLY £289

MICRO-PERIPHERALS DISK DRIVE SPECIALLY DESIGNED TO MATCH QL

The High-Speed disc drive plus interface for the QL.
(See April 'QL USER')

FAST 3.5in Single Disk drive	A MASSIVE 720K Storage (.72Mb)
Double sided, 80 Track	With FREE UTILITIES disc
Completely Boxed	With internal Power Supply Unit

This disk drive complete with the necessary interface
ONLY £295
The second Disk drive can simply be plugged in. £165

Limited offer = PRINTER + (FIRST DISK DRIVE + INTERFACE) = £569
Box of 10 diskettes for above = £55
Prices include delivery, VAT, insurance and cables. All peripherals READY-TO-GO
There are NO HIDDEN EXTRA COSTS with our prices

Cheques/Money Orders to:

BUSINESS MICROS & COMMMUNICATIONS LTD.
18 Oswald Street, Millfield, Sunderland.
Tel: (0783) 674944

NEW — FROM TALENT!

GRAPHIQL

Create superb colour pictures on your QL with TALENT's outstanding new graphics package. It's supplied on two microdrives — the first holds the master program and a printer dump utility, the second, three demonstration pictures. Backup copies can be made. GRAPHIQL comes with a detailed, clearly-written instruction manual, outlining the program's many facilities.

Features include:

- Freehand drawing, 8 colours, optional flash
- Rubber banding, rubber boxes, even rubber circles and ellipses
- Variable size texture definition
- Doodle pad
- Colour and texture fill of any shaped area
- User-definable paint brush — any colour or width
- Colour list for full colour control
- Re-colour facility
- Magnification with panning
- Mirroring and rotation of blocks of screen
- Air-brush effect
- On-line 'help' facility
- Full file-store access
- Printer dump utility

Text can be included in pictures. The characters can be single or double height with flash and underline.

GRAPHIQL pictures can be put into BASIC or assembler programs with the sample routines provided.

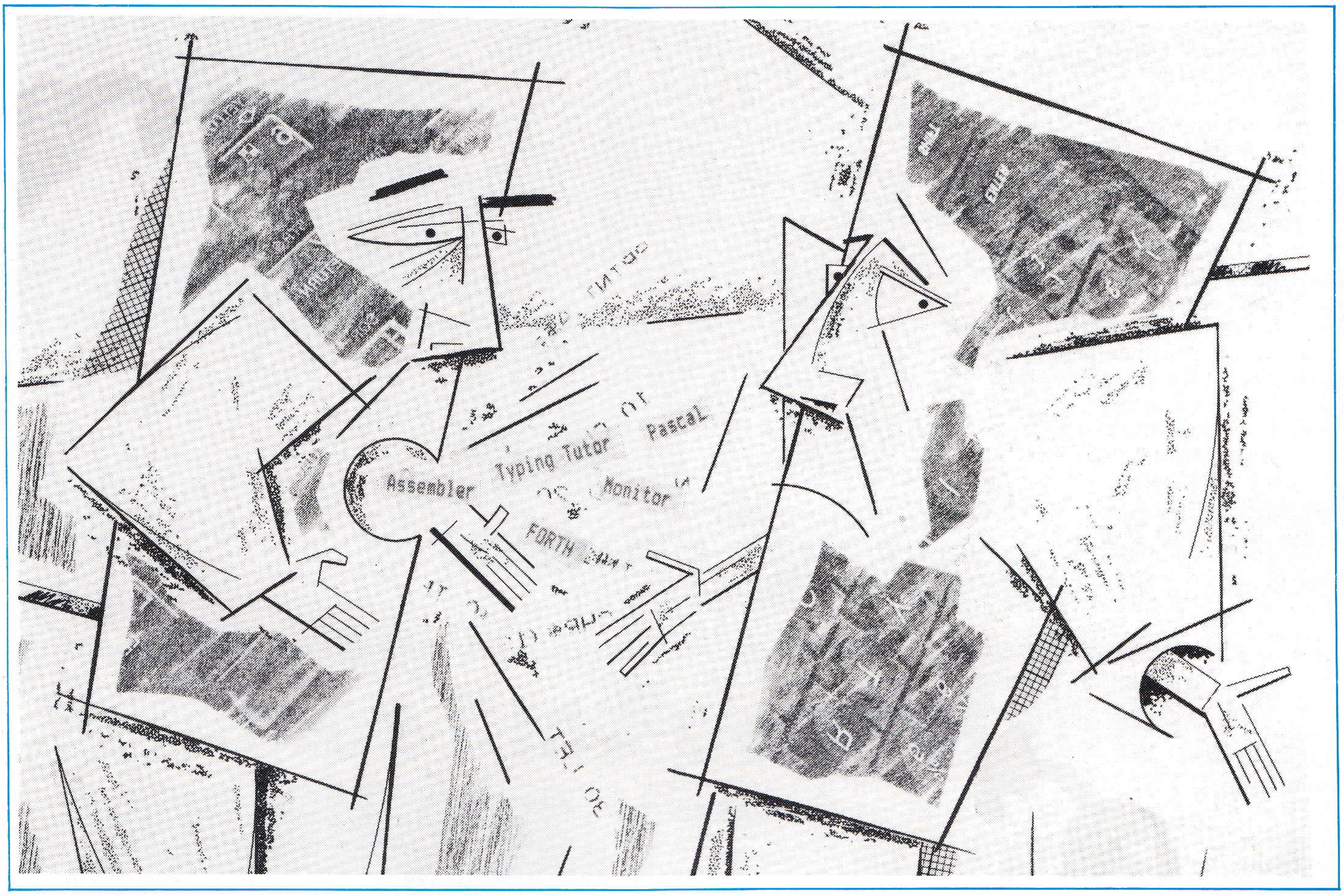
Available in March by mail order direct from:

£34.95
+ 50p postage & packing

QL is a registered trademark of Sinclair Research

TALENT
COMPUTER SYSTEMS
Curran Building,
101 St. James Road, Glasgow G4 0NS
Tel: 041-552 2128 (24-hour Credit Card Hot-Line)
SOFTWARE FROM SCOTLAND

CATEGORICAL COMPUTING



Your chance to win one of ten great prizes from Computer One's extensive range of language/utility packages.

THE PRIZES: The ten winners will each receive one package of their choice from Computer One's current range, including Pascal, FORTH, Assembler, M/C Monitor and Typing Tutor.

TO ENTER: Study the list of six language attributes below and decide which four would be the most important in the design of a new general purpose language for the QL (to replace SuperBasic).

Once you've decided on the four attributes, put them in order of priority (most important first) and then write your answer on a postcard in the same format as the example below (only one ordering is allowed):

eg 1st – D
2nd – A
3rd – F
4th – B

List of Attributes

A.	**Able to accept user command extensions.**
B.	**Program lines easy to understand and follow.**
C.	**Comprehensive debugging and error reporting.**
D.	**Efficient use of memory.**
E.	**Able to integrate system routines.**
F.	**Direct control of external devices.**

RULES: Answers must be written on a postcard only and in the format shown above. Only one entry per person or one entry from any one address. All entries must be sent to:

QL User
C C Competition
Priory Court
30/32 Farringdon Lane
London EC1R 3AU

CLOSING DATE: Entries must be received on or before 20th May 1985.

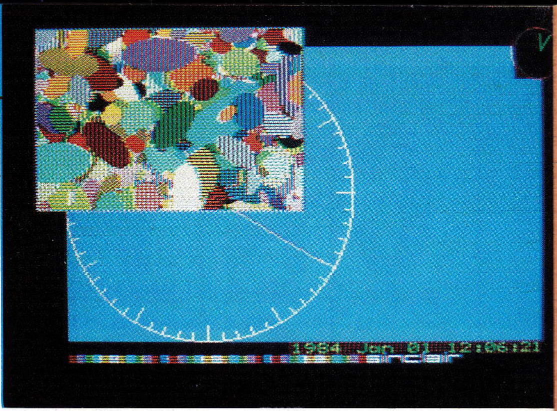
JUDGING: The selection of attributes and final order will be chosen by the Editor. The winners will then be the first ten entries 'out of the hat' with the correct list of attributes and in the right order. Should there be less than ten correct entries, the remaining prizes will go to the first entries chosen with the correct attributes but in the wrong order.

The Editor's decision as to choice and ordering of attributes is final, as is the selection of the ten winners; though their names will be published in a future edition of QL User.

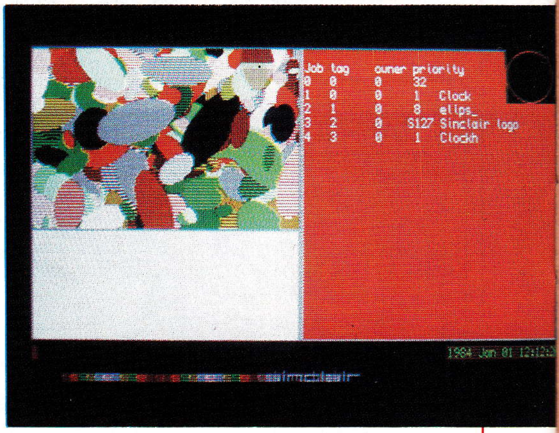
No employee of EMAP or its associate companies, nor anyone connected with Computer One Limited may enter this competition.

QL TOOLKIT

Paolo Baccanello investigates Sinclair's latest software development kit.



```
1984 Jan 01 12:06:21
sinclair
```



```
Job tag   owner priority
0   0     0     32
1   0     0     1   Clock
2   1     0     8   elips_
3   2     0     S127 Sinclair logo
4   3     0     1   Clockh
1984 Jan 01 12:12:2
sinclair
```

Written by Tony Tebby (Qjump) and marketed by Sinclair Research, the QL Toolkit takes over from where SuperBasic leaves off. These extensions to SuperBasic are contained in a 9K machine code file called EXTEN_BIN and are loaded in when the package is booted up.

There is a full screen SuperBasic editor (ED) with the usual diagnostics operating on channel #2. Easy to use and with facilities to undo amendments and page through listings, it is a considerable improvement on QL's primitive line editor. Then there's a file spooler (SPL) which allows you to print out documents or listings in the background so that work may continue uninterrupted – invaluable if you have a busy work schedule.

With much to offer in terms of file handling the kit more than makes up for the lack of built-in commands in SuperBasic. On the level of file maintenance there are the simple commands RENAME, TRUNCATE, VIEW, STAT, WDIR, WSTAT, WDEL, WDEL_F. The first three commands are self evident. The remainder need some explanation. Those beginning with W (wild card) permit files with similar prefixes to be listed (DIR) or deleted (DEL) as a single group. The postfix _F indicates that no confirmation is required. The STAT commands will display not only a file's (or medium's) name but also its length and the date it was last amended.

As far as file I/O is concerned a number of new variants to the OPEN command have been introduced. They are FOPEN, FOP_IN, FOP_NEW, FOP_OVER and FOP_DIR. Only the last two have no direct parallel in SuperBasic. FOP_OVER allows you to open a new file or overwrite an existing file and, FOP_DIR to open a directory. However, what distinguishes all the new commands is that if an operation fails they return an error code and do not crash the program.

Extra functions include FLEN, FTYP, FDAT which return length, type and size of data space.

"With much to offer in terms of file handling, the kit makes up for the lack of built-in commands in SuperBasic."

In the more complex field of Random Access the kit introduces the commands BGET, BPUT, GET and PUT which enable you to move bytes to and from specific locations in files. With the latter two commands data is manipulated in the QL's internal format. Combined with the function FPOS which returns the position of the file pointer these features should appeal to those interested in creating their own databases.

Moving into the area of program execution the emphasis is on machine code and multitasking. At the lowest level two routines are provided which permit you to set device defaults for program and data files (PROG_USE, DATA_USE). Using these, programs may be executed and files accessed by name alone (ie, omitting mdv1,flp1).

Real innovation, however, is to be found in the EX and EW commands introduced to replace EXEC and EXEC_W. These commands may be used to run a whole chain of interlocking machine code programs where each one 'filters' a 'stream' of data and then 'pipes' it onto the next. Those familiar with Unix Shell script may recognise the technique. To aid in the construction of such programs an ET command is also included which permits the chain to be set up but jobs not activated, so that they may be stepped through using a debugger. A number of these filter programs have been provided with the package. The table below shows what each program does.

UC . . . converts input into upper case output (source code supplied)
LNO . . . adds line numbers to input
MORE . . . pages input out on screen
CPY . . . copies input to one or more files
CCT . . . concatenates (joins together) one or more files
PAGE . . . similar to CCT but also pages the output (source supplied)

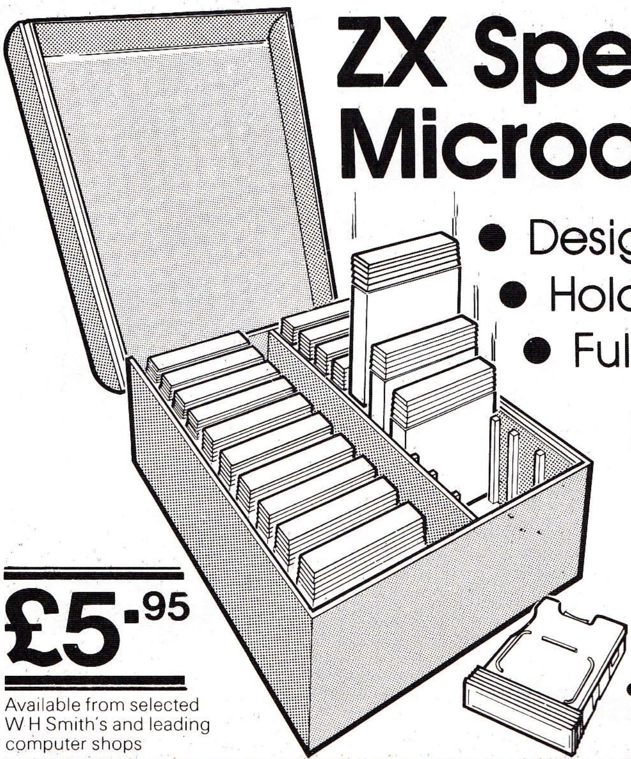
In the area of Job Control, the Toolkit provides a number of much needed routines. The command JOBS will display all programs multitasking along with their tag, priority, status and ownership. Additionally AJOB, SPJOB and RJOB permit you to activate, suspend or terminate specific jobs. Finally JOB$, OJOB, PJOB return a job's name, ID and priority whilst NXJOB allows you to scan the job hierarchy.

With free memory on the QL constantly shifting, extra commands have been provided to supplement the somewhat overworked RESPR routine. ALCHP and RECHP will reserve and release areas on the common heap and CLCHP will clear it altogether. Functions BIN, HEX, BIN$ and HEX$ should warm the hearts of many aspiring machine coders. These allow instant decimal to binary/hex conversions and vice versa. Also, for those developing business software there's FDEC$, IDEC$ and CDEC$ which enable programs to be written efficiently using integer variables. CURSEN and CURDIS enable and disable the screen cursor. WTV and WMON reinstate TV or Monitor windows respectively and more interestingly, CHAR_USE and CHAR_INC allow you to switch character founts and alter character spacing. Last of all, PARTYP and PARUSE are functions to return procedure parameter types and usage.

So, what do we make of the QL Toolkit? Well, the answer lies in how invaluable the 58 or so additions to SuperBasic may be considered to be. For computer novices or business users the package holds little promise; SuperBasic though extended is made no easier. For the hacker or computer buff, however, the package will be most welcome, provided, of course, he hasn't written half these routines himself.

Price: £24.95
Available W H Smiths and leading computer stores or Sinclair direct.

TRANSFORM LTD. Computer Accessories



ZX Spectrum and QL Microdrive Storage Box

- Designed to match QL and Spectrum
- Holds 20 cartridges and index cards
- Fully interlocking

£5.95

Available from selected W H Smith's and leading computer shops

Available Now

QL Centronics Interface	£39.95
QL Dust Cover	£5.00
QL RS 232 Lead	£10.00
Spectrum Monitor Connector	£11.50

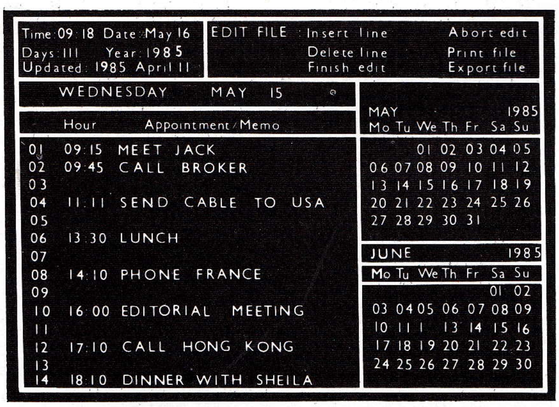
•We also supply printers, monitors, labels, ribbon and listing paper.

Access

TRANSFORM LTD. (Dept. QL) 089 283 4783
'Swatlands', Lucks Lane, Paddock Wood, Kent TN12 6QL.

VISA

Enhance the Power of the QL



```
Time 09:18 Date May 16     EDIT FILE : Insert line      Abort edit
Days:111    Year:1985                  Delete line      Print file
Updated: 1985 April 11                 Finish edit      Export file

   WEDNESDAY      MAY   15
                                            MAY                 1985
   Hour       Appointment/Memo              Mo Tu We Th Fr Sa Su
01  09:15  MEET JACK                              01 02 03 04 05
02  09:45  CALL  BROKER                     06 07 08 09 10 11 12
03                                          13 14 15 16 17 18 19
04  11:11  SEND  CABLE  TO  USA             20 21 22 23 24 25 26
05                                          27 28 29 30 31
06  13:30  LUNCH
07                                          JUNE                1985
08  14:10  PHONE  FRANCE                    Mo Tu We Th Fr Sa Su
09                                                         01 02
10  16:00  EDITORIAL  MEETING               03 04 05 06 07 08 09
11                                          10 11 1  13 14 15 16
12  17:10  CALL  HONG  KONG                 17 18 19 20 21 22 23
13                                          24 25 26 27 28 29 30
14  18:10  DINNER  WITH  SHEILA
```

AGENDA The program that does everything your pocket or desk diary will not do. Agenda is a calendar-based program to remind you of appointments, phone messages, birthdays, holidays, bills and many more.

★ Make a print-out of any display
★ Alarm facility to warn you of an appointment
★ Real timé clock
★ Facility to export to Quill
★ Zoom in to any day facility
★ and more

Introductory offer £19.95 (fully inclusive price) delivered in the U.K. Offer expires 25 May 1985

Q-SOFT

P.O. Box 90
Barnet
Hertfordshire
EN5 5RN
Tel 01-449 7417

Phone or post this coupon. I enclose a cheque for £________
Debit my Access/Visa Acc. No. Card expiry date:________
□□□□□□□□□□□□□□□□
Name:________
Address:________
________ Postcode:________
Tel. No.:________ Signature:________

Fully inclusive price delivered in the U.K. Europe add £1.00. Allow 14 days for delivery

Dealer enquiries welcome.

New

SHADOW SOFT

AREA RADAR CONTROLLER

PRICE
£10.95
includes VAT, postage and packing

SINCLAIR
for the **QL**

Please send me AREA RADAR CONTROLLER on microdrive cartridge for the QL

Name ....

Address ....

....

QLU4

....

Please make cheque or postal order payable to Shadow Soft and send to:
Shadow Soft, 70 Gooseacre, Cheddington, Near Leighton Buzzard, Beds. 0296 668740

A fascinating game and simulation. Control up to 69 aircraft in a busy air traffic zone which includes 2 airfields and 9 airways. Don't run out of fuel and avoid collisions.

VERY ADDICTIVE — HIGHLY FRUSTRATING

QL PROGRAMMERS. *Want someone to market your game? — send it to us for assessment; you will not be ripped off. At worst it will be returned uncopied, at best you will be offered a contract at good royalty rates.*

DIY Assembler

Giles Todd

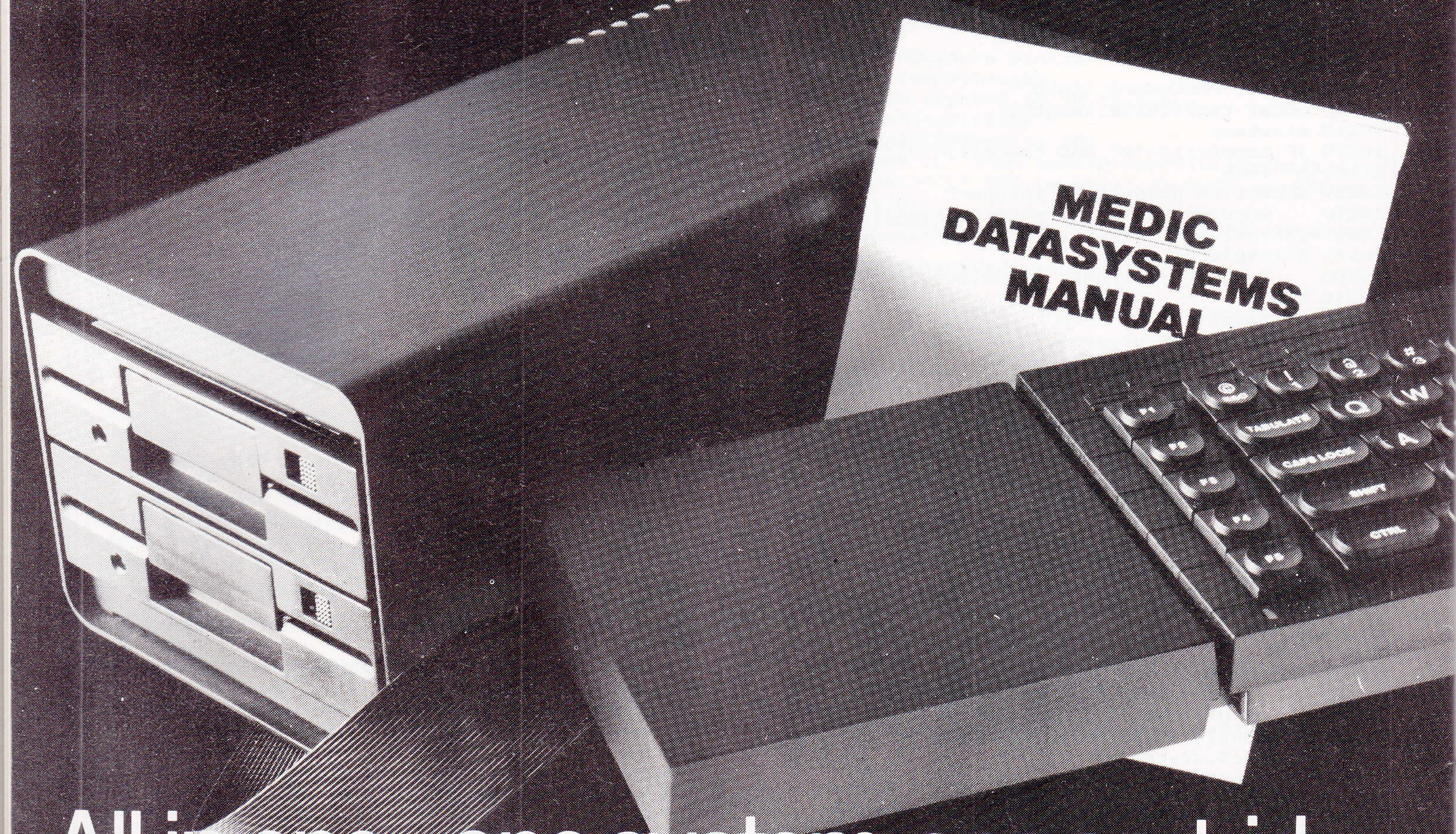
Much of the QL's power is hidden away and accessible only through machine code. To unleash it you not only need a working knowledge of various coding techniques but also a way to delve into the inner workings of QDOS and the 68008 processor. Close to completion, our type-it-in-yourself assembler complements our Machine Code Tutorial providing the means to put theory into practice. Written entirely in SuperBasic (and for that reason it's a little slow), it supports the complete 68008 instruction set as well as the directives ORG, END, EQU, DC and DS.

```
6950 END IF
6960 END DEFine print_object
6970 :
6980 DEFine FuNction cvs$(number,length)
6990 LOCal byte(3),temp$
7000 byte(0)=INT(number/2^24)
7010 byte(1)=INT((number-(byte(0))*2^24)/2^16)
7020 byte(2)=INT((number-(byte(0)*2^24)-(byte(1)*2
^16))/2^8)
7030 byte(3)=INT(number-(byte(0)*2^24)-(byte(1)*2^
16)-(byte(2)*2^8))
7040 temp$=""
7050 temp$=temp$&CHR$(byte(0))
7060 temp$=temp$&CHR$(byte(1))
7070 temp$=temp$&CHR$(byte(2))
7080 temp$=temp$&CHR$(byte(3))
7090 RETurn temp$((5-length) TO 4)
7100 END DEFine cvs$
7110 :
7120 DEFine PROCedure evaluate(mn$,src$,dest$,link
,stype,dtype)
7130 SELect ON link
7140    =0
7150       link0
7160    =1
7170       link1
7180    =2
7190       link2
7200    =3
7210       link3
7220    =4
7230       link4
7240    =5
7250       IF mn$(LEN
(mn$)-1 TO)=".S" THEN mn$
="BT .S":ELSE mn$="BT"
7260       link4
7270    =6
7280       link6
7290    =7
7300       link7
7310    =8
7320       link8
7330    =9
7340       link9
7350    =10
7360       link10
7370    =11
7380       link11
7390    =12
7400       link12
7410    =13
7420       link13
7430    =14
7440       link14
7450    =15
7460       link15
7470    =16
7480       link16
7490    =17
7500       link12
7510    =18
7520       link18
7530    =19
7540       link0
7550    =20
7560       link20
7570    =21
7580       link21
7590    =22
7600       IF mn$="DBRA"
 THEN mn$="DBF"
7610       link4
7620    =23
7630       link23
7640    =24
7650       link24
7660    =25
7670       link25
7680    =26
7690       link26
7700    =27
7710       link27
7720    =28
7730       link28
7740    =29
7750       link29
7760    =30
7770       link30
7780    =31
7790       link31
7800    =32
7810       link32
7820    =33
7830       link33
7840    =34
7850       link34
7860    =35
7870       link35
7880    =36
7890       link36
7900    =37
7910       link37
7920    =38
7930       link38
7940    =39
7950       link39
7960    =40
7970       link40
7980    =41
7990       link41
8000    =42
8010       link42
8020    =43
8030       link43
8040    =44
8050       link44
8060    =45
8070       link45
8080    =46
8090       link46
8100    =47
8110       link47
8120    =48
8130       link48
8140    =49
8150       link49
8160    =50
8170       link50
8180    =51
8190       link51
8200    =52
8210       link52
8220    =53
8230       link53
8240    =54
8250       link54
8260    =55
8270       link55
8280 END SELect
8290 END DEFine evaluate
8300 :
8310 DEFine PROCedure link14
8320 LOCal i,temp$,length
8330 IF dest$="CCR" OR src$="SR" OR dest$="SR" OR
src$="USP" OR dest$="USP" THEN status:RETurn
8340 IF mn$(LEN(mn$))="Q" THEN
8350    IF dtype<>1 THEN operr
8360    word(0)=shift(14)+shift(13)+shift(12)
8370    word(0)=word(0)+reg(dest$,dtype)*shift(9)
8380    word(0)=word(0)+(INT(eval(src$))&&255)
8390    length=2
8400 ELSE
8410 SELect ON dtype=10 TO 16:operr
8420 word(0)=0
8430 temp$=mn$(LEN(mn$)-1 TO)
8440 IF temp$=".B" THEN
8445 IF stype=2 THEN operr
8450    word(0)=word(0)+shift(12)
8460 END IF
8470 IF temp$=".L" THEN
8480    word(0)=word(0)+shift(13)
8490 END IF
8500 IF temp$<>".L" AND temp$<>".B" THEN
8510    word(0)=word(0)+shift(12)+shift(13)
8520 END IF
8530 word(0)=word(0)+amode(dtype)*shift(6)
8540 word(0)=word(0)+reg(dest$,dtype)*shift(9)
8550 word(0)=word(0)+amode(stype)*shift(3)
8560 word(0)=word(0)+reg(src$,stype)
8570 length=2+overhead(stype,mn$,source$)+overhead
(dtype,mn$,destination$)
8580 END IF
8590 IF mn$(LEN(mn$))<>"Q" THEN opcode:ELSE object
$=cvs$(word(0),2)
8600 program_counter=program_counter+length:REMark
 poke the code here
8610 END DEFine link14
8620 :
8630 DEFine FuNction reg(r$,type)
8640 LOCal i,j,temp$,notfound
8650 IF type=13 THEN RETurn 4
8660 IF type>7 THEN RETurn type-8
8670 IF type=7 THEN i=r$(("(A" INSTR r$)+2):RETurn
 i
8690    notfound=-1
8700    FOR j=0 TO 7
8710    IF "A"&j INSTR r$ OR "D"&j INSTR r$ THEN
8720       notfound=j
8730       EXIT j
8740    END IF
8750    END FOR j
8760    RETurn notfound
8780 END DEFine reg
8790 :
8800 DEFine FuNction amode(type)
8810 IF type>7 THEN
8820    RETurn 7
8830 ELSE
8840    RETurn type-1
8850 END IF
8860 END DEFine amode
8870 :
8880 DEFine PROCedure opcode
8890 LOCal a$
8900 object$=cvs$(word(0),2)
8910 IF stype=8 OR (stype=12 AND NOT "Q" INSTR mn$
) THEN word(1)=eval(src$):object$=object$&cvs$(wor
d(1),2)
8920 IF stype=9 OR (stype=13 AND NOT "Q" INSTR mn$
) THEN i=eval(src$):word(1)=INT(i/65536):word(2)=(
i/65536-INT(i/65536))*65536:object$=object$&cvs$(w
ord(1),2)&cvs$(word(2),2)
8930 IF stype=7 OR stype=11 THEN
8940    word(1)=(eval(src$(1 TO ("(" INSTR src$)-1
))&&255)+reg(src$((","  INSTR src$)+1 TO (","  INSTR
 src$)+2),0)*shift(12)
8950     IF ".L" INSTR src$ THEN word(1)=word(1)+sh
ift(11)
8960     IF ",A" INSTR src$ THEN word(1)=word(1)+sh
ift(15)
8970     object$=object$&cvs$(word(1),2)
8980 END IF
8990 IF stype=6 THEN word(1)=eval(src$(1 TO ("(" I
NSTR src$)-1)):object$=object$&cvs$(word(1),2)
9000 IF stype=10 THEN pcrel(src$):object$=object$&
cvs$(word(1),2)
9010 IF dest$<>"" THEN IF dtype=8 OR dtype=12 THEN
 word(1)=eval(dest$):object$=object$&cvs$(word(1),
2)
9020 IF dest$<>"" THEN IF dtype=9 OR dtype=13 THEN
 i=eval(dest$):word(1)=INT(i/65536):word(2)=(i/655
36-INT(i/65536))*65536:object$=object$&cvs$(word(1
),2)&cvs$(word(2),2)
9030 IF dtype=7 THEN
9040     word(1)=(eval(dest$(1 TO ("(" INSTR dest$)
-1))&&255)+reg(dest$(("," INSTR dest$)+1 TO ("," I
NSTR dest$)+2),0)*shift(12)
9050     IF ".L" INSTR dest$ THEN word(1)=word(1)+s
hift(11)
9060     IF ",A" INSTR dest$ THEN word(1)=word(1)+s
hift(15)
9070     object$=object$&cvs$(word(1),2)
9080 END IF
9090 IF dtype=6 THEN word(1)=eval(dest$(1 TO ("("
INSTR dest$)-1)):object$=object$&cvs$(word(1),2)
9100 IF dtype=10 THEN pcrel(dest$):object$=object$
&cvs$(word(1),2)
9110 END DEFine opcode
9120 :
9130 DEFine PROCedure pcrel(op$)
9140 LOCal temp
9150 IF op$(1)="#" THEN
9160     temp=eval(op$(1 TO ("(" INSTR op$)-1))
9170 ELSE
9180     temp=eval(op$(1 TO ("(" INSTR op$)-1))-(pr
ogram_counter+2)
9190 END IF
9195 IF temp<-32768 OR temp>32767 THEN operr
9200 word(1)=temp
9210 END DEFine pcrel
9220 :
9230 DEFine PROCedure link4
9240 LOCal short,temp$,pc,disp,i
9250 short=0
9260 temp$=mn$
9270 IF mn$(LEN(mn$)-1 TO)=".S" THEN
9280     short=1
9290     temp$=mn$(1 TO LEN(mn$)-2)
9300 END IF
9310 word(0)=6*shift(12):IF mn$(1)="D" THEN word(0
)=shift(14)+shift(12)+shift(7)+shift(6)+shift(3)+r
eg(src$,stype)
9320 IF temp$(1)="D" THEN temp$=temp$(2 TO)
9330 FOR i=0 TO 15
9340 IF condition$(i)=temp$(2 TO) THEN
9350     word(0)=word(0)+cond%(i)*shift(8)
9360     EXIT i
9370 END IF
9380 END FOR i
9390 pc=program_counter
9400 IF short OR mn$(1)="D" THEN
9410     pc=pc+2
9420 ELSE
9430     pc=pc+4
9440 END IF
9450 disp=eval(src$)-(program_counter+2):IF mn$(1)
="D" THEN disp=eval(dest$)-pc
9460 IF disp>=-128 AND disp<0 AND mn$(1)<>"D" THEN
 short=1
9470 IF short THEN
9480     pc=program_counter+2
9490     IF disp<0 THEN
9500         word(0)=word(0)+256+disp
9510     ELSE
9520         word(0)=word(0)+disp
9530     END IF
9540 ELSE
9550     word(1)=disp
9560 END IF
9565 IF short=1 AND (disp<-128 OR disp>127) THEN o
perr
9570 object$=cvs$(word(0),2):IF NOT short THEN obj
ect$=object$&cvs$(word(1),2)
9580 IF mn$(1)="D" THEN pc=pc+2
9590 program_counter=pc
9600 END DEFine link4
9610 :
9620 DEFine PROCedure link38
9630 LOCal i,temp$,length
9640 SELect ON stype=2,10 TO 16:operr
9650 word(0)=74*shift(8)
9660 temp$=mn$(LEN(mn$)-1 TO)
9670 IF temp$=".L" THEN word(0)=word(0)+2*shift(6)
9680 IF temp$<>".L" AND temp$<>".B" THEN word(0)=w
ord(0)+shift(6)
9690 word(0)=word(0)+amode(stype)*shift(3)+reg(src
$,stype)
9700 length=2+overhead(stype,mn$,src$)
9710 opcode
9720 program_counter=program_counter+length
9730 END DEFine link38
9740 :
9750 DEFine PROCedure link0
9760 LOCal i,temp$,length,size
9770 sizetemp
9780 IF temp$(i)<>"Q" AND temp$(i)<>"I" AND temp$(
i)<>"X" AND temp$(i)<>"A" THEN
9790     word(0)=shift(15)+shift(14)+shift(12)+size
*shift(6):IF mn$(1 TO 3)="SUB" THEN word(0)=word(0
)-shift(14)
9800     IF stype=1 AND dtype<>1 AND dtype<>2 THEN
9810         SELect ON dtype=2,10 TO 16:operr
9820         word(0)=word(0)+reg(src$,stype)*shift(9
)+shift(8)+amode(dtype)*shift(3)+reg(dest$,dtype)
9830     ELSE
9840         SELect ON dtype=3 TO 16:operr
9850         word(0)=word(0)+reg(dest$,dtype)*shift(
9)+amode(stype)*shift(3)+reg(src$,stype)
9860     END IF
9870     length=2+overhead(stype,mn$,scr$)+overhead
(dtype,mn$)
9880 END IF
9890 IF temp$(i)="A" THEN
9900     SELect ON dtype=1,3 TO 16:operr
9910     word(0)=shift(15)+shift(14)+shift(12)+shif
t(7)+shift(6)+reg(dest$,dtype)*shift(9)+amode(styp
e)*shift(3)+reg(src$,stype):IF mn$(1 TO 3)="SUB" T
HEN word(0)=word(0)-shift(14)
9920     IF size=2 THEN word(0)=word(0)+shift(8)
9930     length=2+overhead(stype,mn$,src$)
9940 END IF
9950 IF temp$(i)="I" OR ((stype=12 OR stype=13) AN
D temp$(i)<>"Q" AND temp$(i)<>"A") THEN
9960     SELect ON stype=1 TO 11,14 TO 16:operr
9970     SELect ON dtype=2,10 TO 16:operr
9980     word(0)=shift(10)+shift(9)+size*shift(6)+a
mode(dtype)*shift(3)+reg(dest$,dtype):IF mn$(1 TO
3)="SUB" THEN word(0)=word(0)-shift(9)
9990     length=2+overhead(stype,mn$,src$)+overhead
(dtype,mn$)
10000 END IF
10010 IF temp$(i)="Q" THEN
10020     SELect ON stype=1 TO 11,14 TO 16:operr
10030     SELect ON dtype=10 TO 16:operr
10040     length=eval(src$):IF length=8 THEN length
=0
10050     word(0)=shift(14)+shift(12)+length*shift(
9)+size*shift(6)+amode(dtype)*shift(3)+reg(dest$,d
type):IF mn$(1 TO 3)="SUB" THEN word(0)=word(0)+sh
ift(8)
10060     length=2+overhead(dtype,mn$)
10070 END IF
10080 IF temp$(i)="X" THEN
10090     IF NOT (stype=1 AND dtype=1) OR NOT (styp
e=5 AND dtype=5) THEN operr
10100     word(0)=shift(15)+shift(14)+shift(12)+shi
ft(8)+reg(dest$,dtype)*shift(9)+size*shift(6)+reg(
src$,stype):IF mn$(1 TO 3)="SUB" THEN word(0)=word
(0)-shift(14)
10110     IF stype=5 AND dtype=5 THEN word(0)=word(
0)+shift(3)
10120     length=2
10130 END IF
10140 IF length=2 THEN object$=cvs$(word(0),2):ELS
E opcode
```

```
10150 program_counter=program_counter+length
10160 END DEFine link0
10170 :
10180 DEFine PROCedure link18
10190 word(0)=hex("4E75")
10200 object$=cvs$(word(0),2)
10210 program_counter=program_counter+2
10220 END DEFine link18
10230 :
10240 DEFine PROCedure link8
10250 LOCal temp$,size,length,i
10260 sizetemp
10270 IF temp$(i)<>"A" AND temp$(i)<>"I" AND temp$
(i)<>"M" THEN
10280 SELect ON dtype=3 TO 16:operr
10290     word(0)=shift(15)+shift(13)+shift(12)+reg
(dest$,dtype)*shift(9)+size*shift(6)+amode(stype)*
shift(3)+reg(src$,stype)
10300     length=2+overhead(stype,mn$,src$)
10310 END IF
10320 IF temp$(i)="A" THEN
10330     SELect ON dtype=1,3 TO 16:operr
10340     word(0)=shift(15)+shift(13)+shift(12)+shi
ft(7)+shift(6)+reg(dest$,dtype)*shift(9)+amode(sty
pe)*shift(3)+reg(src$,stype)
10350     IF size=2 THEN word(0)=word(0)+shift(8)
10360     length=2+overhead(stype,mn$,src$)
10370 END IF
10380 IF temp$(i)="I" OR ((stype=12 OR stype=13) A
ND temp$(i)<>"A") THEN
10390     SELect ON dtype=2,10 TO 16:operr
10400     SELect ON stype=1 TO 11,14 TO 16:operr
10410     word(0)=shift(11)+shift(10)+size*shift(6)
+amode(dtype)*shift(3)+reg(dest$,dtype)
10420     length=2+overhead(stype,mn$,src$)+overhea
d(dtype,mn$)
10430 END IF
10440 IF temp$(i)="M" THEN
10450     IF stype<>4 OR dtype<>4 THEN operr
10460     word(0)=shift(15)+shift(13)+shift(12)+shi
ft(8)+reg(dest$,dtype)*shift(9)+size*shift(6)+shif
t(3)+reg(src$,stype)
10470     length=2
10480 END IF
10490 opcode
10500 program_counter=program_counter+length
10510 END DEFine link8
10520 :
10530 DEFine PROCedure link12
10540 LOCal i,length,size,temp$
10550 sizetemp
10560 IF dest$="" THEN
10570     SELect ON stype=1,2,10 TO 16:operr
10580     word(0)=shift(15)+shift(14)+shift(13)+shi
ft(9)+shift(8)+shift(7)+shift(6)+amode(stype)*shif
t(3)+reg(src$,stype)
10590     IF temp$="ROR" THEN word(0)=word(0)+shift
(10)-shift(8)
10600     length=2+overhead(stype,mn$,src$)
10610     opcode
10620 ELSE
10630     word(0)=shift(15)+shift(14)+shift(13)+shi
ft(8)+shift(3)
10640     IF stype=12 THEN
10650        IF dtype<>1 THEN operr
10660        i=eval(src$)
10670        IF i=8 THEN i=0
10680        word(0)=word(0)+i*shift(9)
10690     ELSE
10700        IF stype<>1 OR dtype<>1 THEN operr
10710        word(0)=word(0)+reg(src$,stype)*shift(
9)+shift(5)
10720     END IF
10730     word(0)=word(0)+size*shift(6)+reg(dest$,d
type)
10740     IF temp$="ROR" THEN word(0)=word(0)-shift
(8)+shift(4)
10750     length=2
10760     object$=cvs$(word(0),2)
10770 END IF
10780 program_counter=program_counter+length
10790 END DEFine link12
10800 :
10810 DEFine PROCedure link21
10820 LOCal i,length,size,temp$
10830 sizetemp
```

```
10840 SELect ON stype=2 TO 11,13 TO 16:operr
10850 SELect ON dtype=2,12 TO 16:operr
10860 IF stype=1 THEN
10870     word(0)=shift(8)+reg(src$,stype)*shift(9)
10880 ELSE
10890     word(0)=shift(11)
10900 END IF
10910 word(0)=word(0)+amode(dtype)*shift(3)+reg(de
st$,dtype)
10920 length=2+overhead(stype,mn$,src$)+overhead(d
type,mn$,dest$)
10930 opcode
10940 program_counter=program_counter+length
10950 END DEFine link21
10960 :
10970 DEFine PROCedure sizetemp
10980 temp$=mn$(LEN(mn$)-1 TO)
10990 IF temp$=".B" THEN size=0
11000 IF temp$=".L" THEN size=2
11010 IF temp$=".W" OR (temp$<>".B" AND temp$<>".L
") THEN size=1
11020 IF temp$=".B" OR temp$=".W" OR temp$=".L" TH
EN temp$=mn$(1 TO LEN(mn$)-2):ELSE temp$=mn$
11030 i=LEN(temp$)
11040 END DEFine sizetemp
11050 :
11060 DEFine PROCedure link41
11070 LOCal i,length,size,temp$
11080 sizetemp
11090 SELect ON stype=2 TO 11,13 TO 16:operr
11100 SELect ON dtype=2,10 TO 16:operr
11110 IF stype=1 THEN
11120     word(0)=shift(8)+shift(7)+shift(6)+reg(sr
c$,stype)*shift(9)
11130 ELSE
11140     word(0)=shift(11)+shift(7)+shift(6)
11150 END IF
11160 word(0)=word(0)+amode(dtype)*shift(3)+reg(de
st$,dtype)
11170 length=2+overhead(stype,mn$,src$)+overhead(d
type,mn$)
11180 opcode
11190 program_counter=program_counter+length
11200 END DEFine link41
11210 :
11220 DEFine PROCedure link30
11230 LOCal i,length,size,temp$
11240 sizetemp
11250 SELect ON stype=2,10 TO 16:operr
11260 word(0)=shift(14)+shift(10)+shift(9)+size*sh
ift(6)+amode(stype)*shift(3)+reg(src$,stype)
11270 length=2+overhead(stype,mn$,src$)
11280 opcode
11290 program_counter=program_counter+length
11300 END DEFine link30
11310 :
11320 DEFine FuNction inside(s$,f$)
11330 IF s$ INSTR f$ THEN RETurn 1:ELSE RETurn 0
11340 END DEFine inside
11350 :
11360 DEFine PROCedure link7
11370 LOCal i,length,size,temp$
11380 sizetemp
11390 SELect ON stype=2,10 TO 16:operr
11400 word(0)=shift(14)+shift(9)+size*shift(6)+amo
de(stype)*shift(3)+reg(src$,stype)
11410 length=2+overhead(stype,mn$,src$)
11420 opcode
11430 program_counter=program_counter+length
11440 END DEFine link7
11450 :
11460 DEFine PROCedure link45
11470 LOCal i,length,size,temp$
11480 IF stype<>1 THEN operr
11490 word(0)=shift(14)+shift(11)
11500 IF ".L" INSTR mn$ THEN
11510     size=3
11520 ELSE
11530     size=2
11540 END IF
11550 word(0)=word(0)+size*shift(6)+reg(src$,stype
)
11560 object$=cvs$(word(0),2)
11570 length=2
11580 program_counter=program_counter+length
11590 END DEFine link45
```

NOW an expansion system as versatile as the QL



All in one - one system, one cartridge

MEDIC takes up where others leave off. Why narrow down your choices? The Q.L. was intended as a versatile computer capable of fulfilling many roles. With our system cartridge, the choice is yours. As well as a disc interface including parallel port you can have any one of the following: memory upto 512K, modem, Eprom card (192K) experimental card including front interface. You can upgrade to any level in line with your needs. Our disc interface affords full Psion, versions 1 and 2, compatibility.

Ask your dealer for details or order direct:

medic datasystems limited

76 Grainger Close, Basingstoke, Hants. RG22 4EA
Telephone 0256 460748

□ Please send me further details on your products.

I wish to order the following: (please tick required items)

Individual Products

□ 64K Memory	£99.95
□ 128K Memory	£129.95
□ 256K Memory	£169.95
□ 512K Memory†	£259.95
□ Eprom card socketed (192K – excluding Eproms) experimentation card with front interface	£24.95
□ Dust Cover that doubles up as a stand	£14.95
□ Modem*	£79.95

†Can only be powered if Medic disc system connected

**Delivery beginning April*

Packages

□ 1 Megabyte disc drive (cased, inc. PSU) disc interface and parallel interface, with cables	£249.96
□ 2 1 Megabyte disc drives, (cased, inc. PSU) disc interface parallel interface, with cable	£399.95

□ 1 megabyte disc drive, memory inc., disc interface and parallel interface.

1 disc drive		2 disc drives
□ 64K	£299.95	□ £449.95
□ 128K	£329.95	□ £479.95
□ 256K	£359.95	□ £509.95
□ 512K	£449.95	□ £595.95

To order products please make cheques/P.O.'s payable to MEDIC DATASYSTEMS LTD. Allow 28 days for delivery. All prices include VAT. Please add £5 p&p.

NAME______________________________

ADDRESS___________________________

____

Signed_____________________________

THE SINCLAIR QL SERIES

Series Editor: Robin Bradbeer
Foreword by Nigel Searle,
Managing Director, Sinclair Research

recommended by "QL User" magazine

at last the complete QL library

THE SINCLAIR QL SERIES
MACHINE CODE PROGRAMMING ON THE SINCLAIR QL
DATABASE MANAGEMENT ON THE SINCLAIR QL
USING GRAPHICS ON THE SINCLAIR QL
Garry Marshall
MAKING THE MOST OF THE SINCLAIR QL
Dick Meadows
PROFITING FROM THE SINCLAIR QL
Barry Miles
INTRODUCING THE SINCLAIR QL
Garry Marshall
INTRODUCTION TO SUPERBASIC ON THE SINCLAIR QL
Dick Meadows
ADVANCED PROGRAMMING WITH THE SINCLAIR QL
Martin Gandoff
DESK TOP COMPUTING WITH THE SINCLAIR QL
Barry Miles
WORD PROCESSING WITH THE SINCLAIR QL
Mike O'Reilly

NEW **Machine Code Programming on the Sinclair QL**
Martin Gandoff

NEW **Database Management on the Sinclair QL**
Mike O'Reilly

NEW **Using Graphics on the Sinclair QL**
Garry Marshall

NEW **Making the Most of the Sinclair QL**
QL SuperBasic and its applications
Dick Meadows

NEW **Profiting from the Sinclair QL**
financial and other spreadsheet applications
Barry Miles

192 pages illustrated £7.95 paperback

fully up-to-date with Psion Systems version 2.00 software

previously published in the series

Introducing the Sinclair QL
Introduction to SuperBasic on the Sinclair QL
Advanced Programming with the Sinclair QL
Word Processing with the Sinclair QL
Desk Top Computing with the Sinclair QL

192 pages illustrated £6.95 paperback

Order now! Just fill in the order form and return to us with your cheque or money order. Your books will be sent POST FREE in the UK, within 28 days.

ORDER FORM

To: Hutchinson Computer Publishing
17-21 Conway Street, London WIP 6JD

Please send me:

...... copies of **Machine Code Programming on the Sinclair QL**	£7.95 each	09 160531 8
...... copies of **Database Management on the Sinclair QL**	£7.95 each	09 160551 2
...... copies of **Using Graphics on the Sinclair QL**	£7.95 each	09 160521 0
...... copies of **Making the most of the Sinclair QL**	£7.95 each	09 160561 X
...... copies of **Profiting from the Sinclair QL**	£7.95 each	09 160541 5
...... copies of **Introducing the Sinclair QL**	£6.95 each	09 158941 X
...... copies of **Introduction to SuperBasic on the Sinclair QL**	£6.95 each	09 158951 7
...... copies of **Advanced Programming with the Sinclair QL**	£6.95 each	09 158961 4
...... copies of **Word Processing with the Sinclair QL**	£6.95 each	09 158971 1
...... copies of **Desk Top Computing with the Sinclair QL**	£6.95 each	09 158981 9

Overseas orders — add £1 p&p per title.

I enclose my cheque/money order made payable to TBS for

£______________________

My name and address is ______________________________

____

____

Signed ______________________

HCP

NEWS BACKGROUND

The stories behind the news and latest product information, compiled by Sid Smith.

The Silicon Disc

The QL's 0.5Mb wafer scale integrated (WSI) memory circuit, due for release before the year end at around £300, will be "the most exciting event in the computer industry during 1985", says Sir Clive.

Addressing a QL seminar at the London Hilton, Sir Clive extolled this "silicon hard disk" as a uniquely fast, rugged, compact, and reliable storage device – ideal for the Spectrum-like Sinclair portable due for release in 1986. An announcement about a new factory to produce the WSI RAM should be made within "a few weeks".

So admirable, indeed, is this 4″ diameter mega-chip, that Clive raised doubts about his commitment to the QL's promised Winchester interface.

"Hard disk interfaces are now available for the QL from other companies, and we feel that the silicon disk might be more appropriate for this kind of computer. If people want greater storage space they can stack the WSI disks."

Nigel Searle, Sinclair Research MD, explained that the battery which provides back-up power for the silicon disk is provided with a low-power warning light, and can be changed while the device stays plugged into the QL's expansion bus – obviating the need to copy data to another storage device.

The computer industry's Holy Grail of wafer scale logic (as opposed to memory) circuits won't be a straightforward development of Sinclair's present technology – which is based on the ideas of renegade British inventor Ivor Catt.

Explained Sir Clive, "Memory lends itself particularly well to the sort of wafer scale arrangement we have" – where the circuit builds itself up across the wafer by testing individual blocks of the circuit and incorporating or rejecting the blocks according to whether they work. So you have to provide the system with testable blocks, and that's easy to do with memory circuits but less easy with logic circuits.

"But we certainly have a head start over other manufacturers. And many of the patents we're filing now have moved away from Ivor Catt's ideas."

The silicon hard disk should also find a place in the Spectrum-derived portable which Sinclair plan to release next year – a new interface is all that's needed to convert the WSI technology for Z80-based architecture. But as to when that happens – it's anyone's guess.

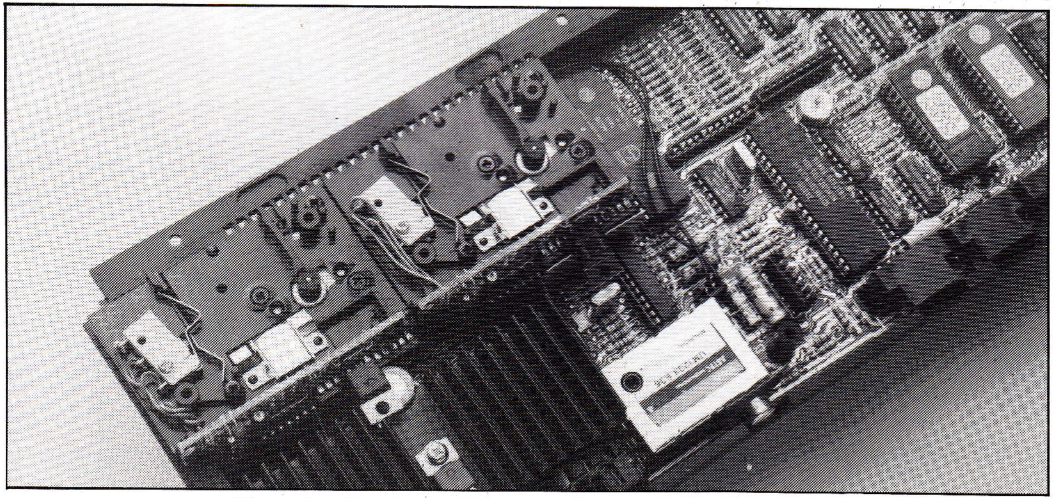
Curiously, we got the impression that Sir Clive is hedging on his previously cast-iron commitment to cathode ray tube technology – the Sinclair flat TV tube.

Is the company working on a larger version of the flat CRT for computer use, we asked him, or would he be using some kind of magnification technique?

"It's a new technology which we can't describe at present because we haven't filed the patents."

But is it a CRT?

"I can't tell you any more than that. It's a light-up display. The view we take is that you've got to have low-power, obviously, but what people want is not dingy old liquid crystal, they really want what they get on a CRT; we're aiming to give them that, but how we're doing it has to be a secret."



The much-maligned microdrive mechanism revealed for all to see.

Microdrive Mechanics

Any QL user whose microdrives fail to give perfect performance can have a new machine – and that's official from Sir Clive.

"I'd be naive to suppose that you didn't know we'd had problems with the microdrives on early issues of the QL.

"This arose because we translated the technology from the Spectrum, where we'd had superb results from microdrives, and made a few changes which didn't work out as well as we'd hoped. But we've sorted all that out now – as is confirmed by the adoption of microdrives by ICL."

Clive rejected our suggestion that there must therefore be QLs in use with inferior microdrives – "faulty machines will have been returned by now," he believed.

The company was aiming for a QL return rate of under 2%, compared with its present 3.5%. Nevertheless, we managed to extract a promise that anyone whose microdrives give "less than 99% performance" could send their QL back to Sinclair Research.

Hunting For Hackers

"Conventional wisdom in the States assumes there aren't any hackers left. I don't believe that's true. I think they've just been obscured by the professional and business users that Apple and IBM sell to. Yes, they're a small percentage of the market, but I think in absolute numbers there are actually more of them than there were two or three years ago."

And so confident is Nigel Searle that these American hackers (in the old-fashioned sense of code junkies rather than comms intruders) are a viable market for the QL that he's off to lead the company's US arm in Boston, Mass.

"There are an enormous number of people in the States who've already got QLs, who've had friends buy them in England and send them over there. And this hacker's market is one which the QL is very well suited to. It's a price-sensitive market, and there aren't many of those in America. The hackers know that they don't *need* a 68000 machine, but they know they *want* one." Wishful thinking? Well who ever thought they needed a ZX81?

It sounds very sensible. Sinclair won't need much advertising (hackers know what they like) nor much dealer support (hackers think they can take care of themselves); it confronts no established US giants (compare Acorn's assault on the American education market) and capitalises on the QL's high spec and low price (though, at $499, its pricing ignores the computer industry's long-established pound/dollar parity).

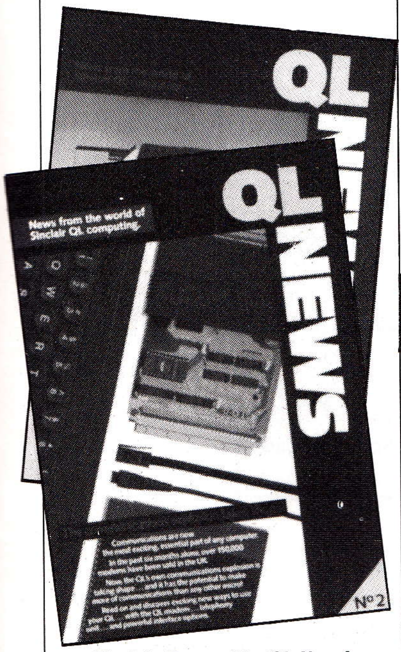
A similarly focussed and economical exercise couples this appeal to the *cognoscenti* with a hardware bundle for US first-time buyers. In a joint marketing effort with American Express; Sinclair will sell the QL, plus printer and monitor, as "a home productivity station" for around $1000.

In short, Sinclair's American campaign will be a modest venture appropriate to inevitably modest results. But if Atari's ST *does* appear then Nigel should come straight home; he'll be needed here!

NEWS BACKGROUND

Promoting Problems

Eating a big chunk of Sinclair's new £0.75m advertising budget for the QL, the multi-page glossy advertising supplement carried in several magazines (our's included) does a great job of promoting the companies featured in its pages – and, apparently, of infuriating everyone else.



Sinclair Research's 'QL News'.

Most strident amongst these bemused outsiders is PCML, who market a QL implementation of Digital Research's CP/M 80 – that doyen of 8-bit operating systems.

Lauding it as "the key to American sales", boasting of its tens of thousands of software packages, vaunting it as "the Amstrad killer", PCML boss John Fuller has no constricting modesty about thè combination of QL and CP/M 80 – nor about his frustration at a "lack of co-operation" from Sinclair Research.

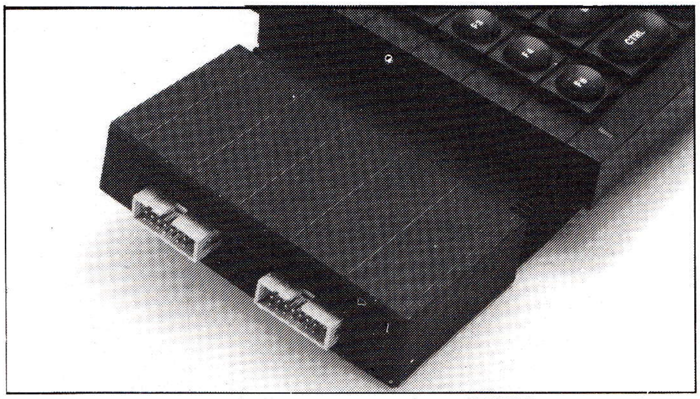
"Buying a licence from Digital to sell CP/M is expensive," says Fuller, "but it does go down with quantity. Licensing just one unit, for instance, would cost £75, but the unit cost comes down to £5 if I buy 10,000 licences. That's still an investment of £50,000; and when you add another £150,000 or £200,000 for the cost of components you can see why we want Sinclair's help to win those mass market sales."

Fuller's mood isn't helped by what he sees as unfair Sinclair assistance to *certain* rivals who market a CP/M 68K operating system.

"Sinclair needs an add-on operating system to sell the QL," he says. "Clive is trying to establish QDOS as an industry standard, but he'll never make it; no software house is going to write a major package for QDOS when they can guarantee themselves a huge market by writing, say, for MS-DOS.

"CP/M 68K is a joke. There's no software for it and there never will be. And the same thing applies to the 68K/OS. Sinclair's only choice is to promote CP/M 80 – but they said they'd only recommend it if we paid them a royalty!

"Sinclair should give us technical help so that there's no clash between our CP/M and their future products. They should recommend CP/M, and put pressure on retailers to carry it. Believe me, it's in their interest as well as ours."



The CP/M 80 interface from QL Plus (PCML).

USER GROUP NEWS

A special forum for news and views from the independent users' group, compiled by Leon Heller, IQLUG Chairman

Around 200 people attended the Swindon workshop and sessions on BCPL, APL, C, Forth and Archive attracted large audiences. Tony (QDOS) Tebby was available to answer questions on QL software and hardware. Several interesting new software and hardware products were launched.

GST have developed a C compiler rejoicing in the name of QC. Although it lacks such niceties as floating point,multi-dimensional arrays and structures, it is quite adequate for systems programming. I plan to port a "smart terminal" program in C for my CP/M system over to the QL, which should be a good test of it.

Metacomco demonstrated their range of QL software, and they hope to have Lattice C available soon, which is a full implementation of the language, and full Cambridge LISP, now that .5Mb RAM expansions are available.

Real Time Systems announced their QL development software – Pascal and C cross-compilers – but we didn't have a VAX so they couldn't demonstrate them!

Dataller had a very impressive package for insurance brokers, written in assembly language, which really used the display features of the QL to advantage.

PCML had a prototype Z80 card for the QL, with Personal CP/M in EPROM, running Wordstar off microdrives. Output to the display was a bit leisurely, but they hope to speed it up.

ICL turned up with a One-per Desk, which attracted some interest as many members work for large companies at which the OPD is aimed.

We also performed some networking experiments, with two standard QLs networked to a disk-based system. After a great deal of trial and error (the QL documentation wasn't a lot of use) we got it all working properly, and even managed to get the disk system operating in "broadcast" mode to the other two. What is needed now is some assembly language software to allow one system to run unattended as a file-server.

Two Into One

One of our members, John Lawlor, has both the CST disk interface, and the Simplex Data .5Mb RAM expansion module. Since the Sinclair expansion unit is not yet available he has built an adaptor that interfaces both units to the expansion port. He tells me that it works perfectly, although in theory the bus signals should be buffered for more than one peripheral device.

Long Strings

Although SuperBasic allows very long strings, the INPUT statement gives a "buffer full" error when strings longer than 128 bytes are input. A couple of IQLUG members wishing to input long strings have asked how to get round this problem, which might be puzzling other users. The solution is quite simple – just use INKEY$ and the string concatenation operator, '&'. The following program fragment illustrates the technique:—

```
long_string$ = ""
REPeat get_string
  k$ = INKEY$(−1)
  IF CODE(k$) = 10 THEN
  EXIT get_string
  PPRINT k$;
  long_string$ = long_
  string$ & k$
END REPeat getstring
:
PRINT
PRINT long_string$
```

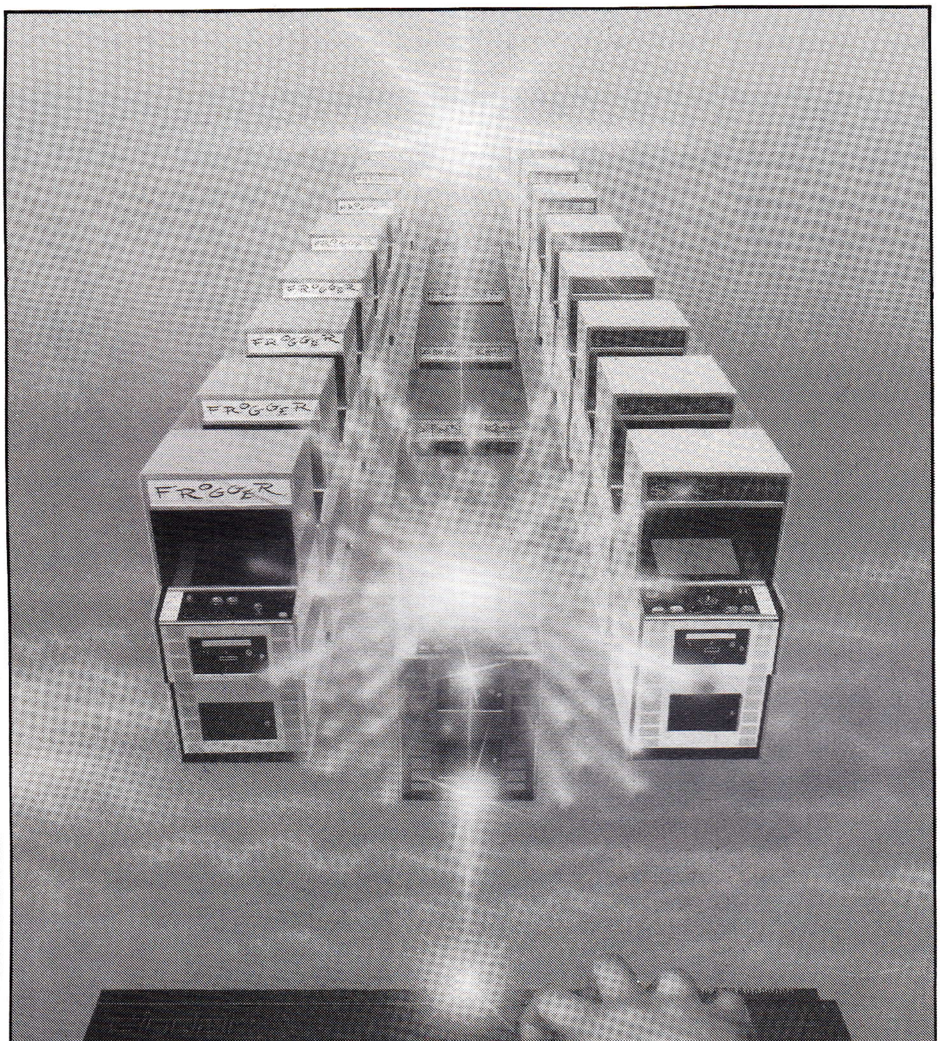
The Independent QL Users' Group (IQLUG) publishes a monthly newsletter, maintains a software library, supports local groups, and provides members with a free advice service. Workshops are arranged from time to time in different parts of the country. Further details are available from:

Brian Pain, 24 Oxford Street, Stony Stratford, Milton Keynes MK11 1JU. Tel: (0908) 564271

GAMES PROGRAMMING

The second part of our exciting introduction to the art of low level coding, by Ian Williams and Steven Hollywood.

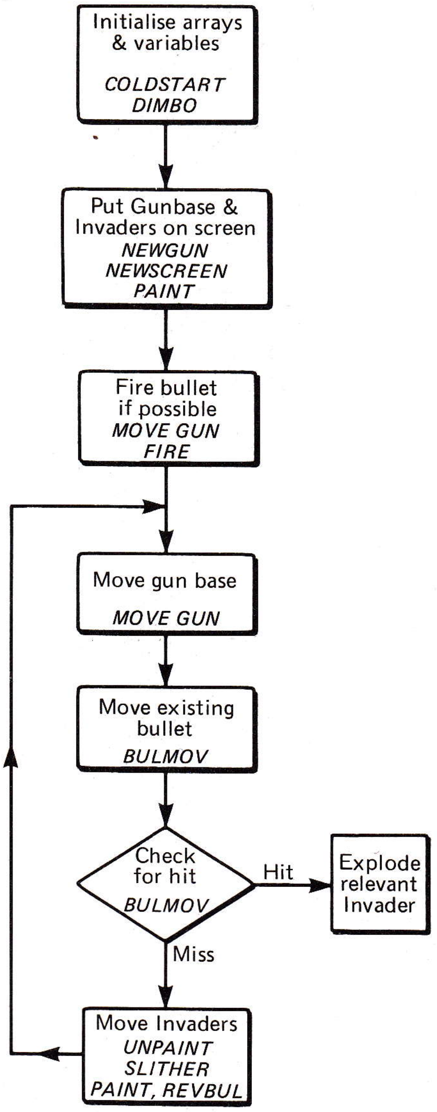
Last month we dealt with the mechanics of designing sprites, putting them on screen and moving them about. To accomplish this we used nine routines, each of which had a specific function and one large routine, 'Coldstart', which appeared with REMs on quite a few lines. It's particularly important that these lines are not 'activated' until we've published the routines to which they refer, otherwise the program will crash. If you've entered last month's listing then you should have a screen full of little invaders just bursting to attack you.



This month we're going to add a gunbase at the bottom of the screen (which will be under joystick or cursor key control), sound effects, both for firing the gun and hitting the invaders and bullets from the gun.

The modular diagram (fig. 1) shows the program so far. From that the sequence of program steps can be seen and the names of this month's new routines.

Fig 1



NB.
Words in *italics* refer to routines in the program

The largest routine is BULMOV and is fully charted in *fig. 2*. Movement of the bullet, reading of its position, unplotting (deleting) of the old bullet and saving the new position are all fairly obvious. d6 and d7 are the registers which hold the x and y postions of the bullet and PLOT is the routine which deletes the bullet by the simple expedient of plotting another one on top using the 'eor' function — a useful way of removing

Fig 2

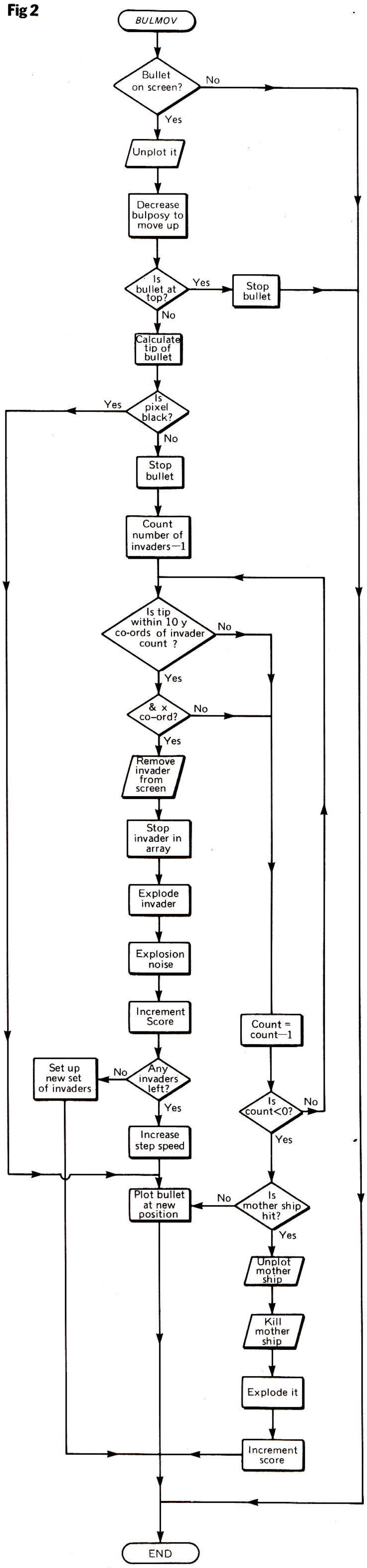
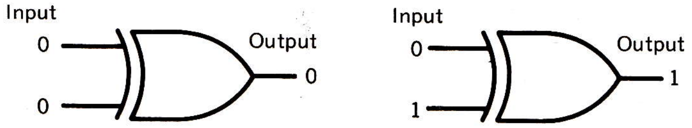


Fig 3



images from the screen.

The EOR gate *(fig. 3)* has two inputs. If both inputs are the same, then the result is a zero. If either input is different then the result is 1. Our screen starts as black, the long word for which is 8 zeros. *Fig. 4* shows a black screen long word being 'EORed' with a pattern, such as a bullet. The result is that the pattern now appears on the screen. If we then 'EOR' the pattern with the same pattern it returns zeros, and the screen reverts to its original state.

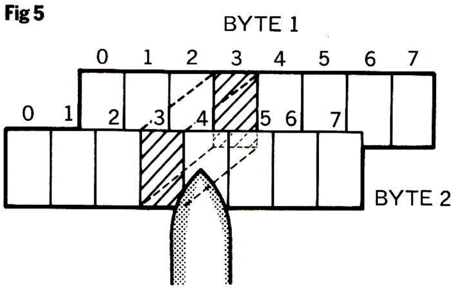
Fig 4

Long Word (black)	:	00000000	$00
Bullet pattern	:	01011010	$5A
Result of EOR	:	01011010	$5A
Bullet pattern	:	01011010	$5A
Result of EOR	:	00000000	$00

Checking to see if a collision between the bullet and anything else has occurred is the next problem. This means we have to read the screen to detect a change in the pixel settings and for this we have a routine called ID_COL.

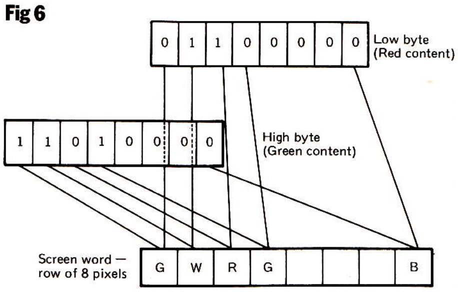
When using ID_COL, the first branch is to 'CALC_ADDR' which is fully explained on the listing. CALC_ADDR then returns with the onscreen address of the word containing the data for the relevant pixel (in this case the one at the tip of the bullet). It also returns (in d2) the position of the bit concerned in each byte of the word pointed to by a1. From last month you'll remember that each pixel is controlled by two bytes. *Figs. 5 & 6* shows this clearly. Here, d2 would contain '3' showing that it is the fourth bit in each byte that's concerned.

Fig 5



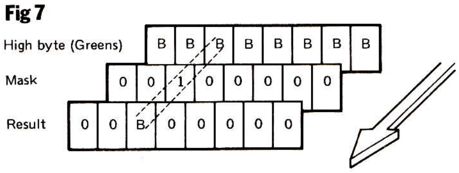
"Each pixel is controlled by 2 bits — one from each byte"

Fig 6



ID_COL now reads 'mask_tab', a table of masks through which only the information concerning the relevant bit can be passed. In effect, 'mask_tab' is placed over the bytes as shown in *fig. 7* allowing the result through from that bit number only. This is done by using the AND function.

Fig 7



The remainder of BUL_MOV is explained in the listing and is fairly straightforward except for the sound. Producing sound effects in machine code means accessing the OS through a TRAP#1 instruction. Sound is produced on the Intel second processor and, as such, takes no main processor time other than the access command.

The sound command uses the same parameters as in BASIC and are as follows:

Pitch (1)	byte	**0 to 255**
Pitch (2)	byte	**0 to 255**
Grad_x (step)	2 bytes	**–32768 to 15**
Duration	2 bytes	**–32768 to 32767**
Grad_y (step)	nybble	**–8 to 7**
Wrap	nybble	**0 to 15**
Random	nybble	**0 to 15**
Fuzzy	nybble	**0 to 15**

To enter a single beep (10000,50) through machine code proceed thus:

Label (Whatever you want to call it)
dc.b 10,8
dc.l $aaaa (any label)
(These values, which must be aligned to a word boundary, form the bytes sent to the OS to inform it that a sound command is following.)
dc.b 51,0 (pitch (1), pitch (2))
(49 is pitch (1) (machine code requires that 1 is added) and 0 is pitch(2))
dc.b 0,0 (LSB first)
dc.b 16,39 (LSB then MSB)
(LSB of 10000 then MSB)
dc.b $00,$00,1 (GRAD, RAD, etc)
(1st nybble = Grad_y, 2nd nybble = wrap, 3rd nybble = Random, 4th nybble = fuzz, 3rd byte = end of data)

To call the sound:

```
 lea      label,a3   ; Tells the OS
                       data location
 moveq #17,d0        ; Command byte
                       for sound
 Trap   #1           ; Execute
```

The MOVE GUN routine involves another OS function, Keyrow, which is the machine code equivalent of the equivalent of the basic function.

The row to be read is passed in d1, processed by the OS and a value returned in the same register.

When adding this month's program to last month's the whole listing, *excluding* the sprite definitions (lines 172–175 should then be added immediately after the last of last month's sprite definitions (sprite 7).

The REMs (;) should be removed from the following lines in last month's listing before assembly:

Lines	Label	Instruction	
22	restart	lea	bulposx,a0
23		move.w	#–1,a0
26		bsr	new_gun
34		bsr	move_gun
39		bsr	bulmov
47		bsr	rev_bul
51		bsr	rev_bul

As before, to enter and run this program after assembling, type:

address = respr (6000)

which will reserve space in memory.

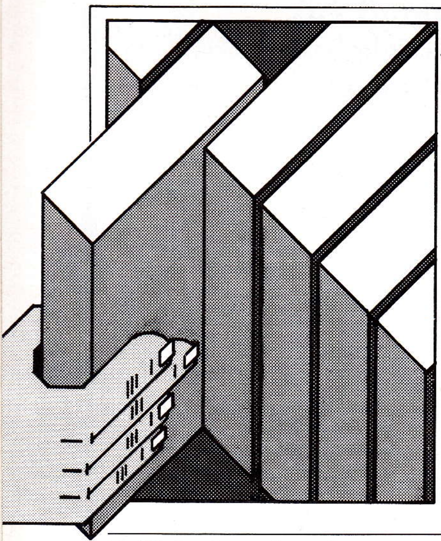
lbytes mdv1_(name)_cde,address will then load the code into memory and 'Call address'.

Next month more procedural enhancements to bring the game up to professional standards.

```
        ****************************************************
        *                                                  *
        *                 << PALADIN >>                    *
        *                                                  *
        *                      by                          *
        *                                                  *
        *  STEVEN HOLLYWOOD        &       IAN G WILLIAMS  *
        *                                                  *
        ****************************************************
13 ; This routine is the m/c equivalent of the basic Keyrow cammand
14 ;
15 keyrow
16          lea     temp,a3         ; Set a3 pointing to the data
17          move.b  d1,6(a3)        ; Stores keyrow row to be read
18          moveq   #17,d0          ; Store keyrow command no.in d0
19          trap    #1              ; Through trap to OS
20          rts
21 ;
22 ; This routine  moves the gunbase across the screen & fires bullets
23 ;
24 move_gun                        ;
25          moveq   #1,d1           ; These lines read keyrow(1) i.e.
26          bsr.s   keyrow          ;  test for a keypress
27          btst    #6,d1           ; This checks for a fire(SB or JStik)
28          beq.s   notfir          ; If not then don't fire
29          bsr.s   fire            ;
30 notfir   lea     gunpos,a0       ; These lines load the gunposition
31          move.w  (a0),d0         ;  into d0
32          move.w  d0,d2           ; Gunpos. saved in d2
33          btst    #1,d1           ; Tests for left movt (cursor or Jstik)
34          beq.s   gnl             ;
35          cmp.w   #3,d0           ; If left test for extremities
36          ble.s   gnl             ; No movmt if extreme left
37          subq.w  #2,d0           ; Moves gun 2 pixels left
38 gnl      btst    #4,d1           ;  These next 5 lines
39          beq.s   gnr             ;  do the same testing
40          cmp.w   #499,d0         ;  but this time for
41          bhi.s   gnr             ;  right movement
42          addq.w  #2,d0           ;
43 gnr      move.w  d0,(a0)         ; Stores new gun position in gunpos
44          moveq   #8,d5           ; Sets up sprite code for a gun
45          move.w  d0,d6           ; Sets d6 to the new position
46          move.b  #240,d7         ; Keeps gun at 240 on y axis
47          bsr     plot            ;
48          move.w  d2,d6           ; These lines erase the
49          bra     plot            ;  old gun image
50
51 ;
52 ; This routine initially positions and draws the gun base
53 ;
54 new_gun                         ;
55          lea     gunpos,a0       ;
56          move.w  #256,d6         ; Position of first gun - x-axis
57          move.w  d6,(a0)         ; Stores position
58          move.b  #240,d7         ; Position of gun - y-axis
59          moveq   #8,d5           ; Sprite number of gunbase
60          bra     plot            ; Branch to plotting routine
61 ;
62 ; Establishes the initial position of a bullet and creates the
63 ; incredibly realistic and versatile sound effects....
64 ;
65 fire
66          lea     bulposx,a0      ; Test to ensure no bullet on screen
67          tst.w   (a0)            ;
68          bpl.s   no_fire         ;
69          move.w  gunpos,d6       ; Copy gunbase pos. into bullet
70          move.w  d6,(a0)         ;  position
71          lea     bulposy,a0      ; These 3 lines
72          move.b  #232,d7         ;   establish the initial
73          move.b  d7,(a0)         ;     y positon of bullet
74          moveq   #9,d5           ; Sprite code
75          bsr     plot            ;
76          lea     zap,a3          ; Sound:see text..
77          moveq   #17,d0          ;
78          trap    #1              ;
79 no_fire rts
80 ;
81 ; This routine moves the bullets towards the descending aliens
82 ;
83 bulmov
84          lea     bulposx,a0      ; Loads x pos. of bullet into d6
85          move.w  (a0),d6         ;
86          bmi.s   no_bul          ; If negative pos.value then no bullet
87          lea     bulposy,a1      ; Loads y pos. of bullet into d7
88          move.b  (a1),d7         ;
89          moveq   #9,d5           ; Sprite code of bullet
90          bsr     plot            ; This branch unplots the old bullet
91          subq.b  #1,d7           ; Moves bullet up one pixel
92          cmp.b   #9,d7           ; Tests to see if top has been reached
93          bhi.s   stilbul         ;
94          move.w  #-1,(a0)        ; If top then cancel bullet
95 no_bul   rts                     ;
```

```
 96 stilbul move.b  d7,(a1)          ; Save new y position in bulposy
 97         addq.w  #3,d6            ; Calculate Tip  of bullet position
 98         bsr     id_col           ; Tests for hit
 99         beq     no_hit           ;
100         move.w  #-1,(a0)         ; If hit then cancel bullet
101 ;        cmp.b   #199,d7          ; Tests for barrier hit
102 ;        bls.s   notbhb           ;
103 ;        and.w   #$ff00,d2        ; Checks to see colour
104 ;        beq     bul_hit_bar     ; This explained in future issue
105 ;        bra     endbomb         ;
106 notbhb  lea     xpos,a0          ; Loads pos. of x and y arrays (invader
107         lea     ypos,a1          ;  positions) into a0 and a1
108         moveq   #num_sp-1,d4     ;
109 nextst  move.b  d7,d5            ; Save hit pos. (y co-ord) in d5
110         sub.b   (a1)+,d5         ; Subtract 1st inv.position from d5
111         cmp.b   #10,d5           ; If Inv. is closer then 10 pixels to
112         bls.s   poss             ; where hit was recorded then poss/hit
113         addq.l  #2,a0            ; Increase y array pointer
114         bra.s   g_next           ;
115 poss    move.w  d6,d0            ; Saves hit pos (x co-ord) in d6
116         sub.w   (a0)+,d0         ; Subtract inv.x pos from d0
117         bpl.s   not_neg          ; These two lines convert d0 (dist from
118         neg.w   d0               ; hit) to a positive value
119 not_neg cmp.w   #10,d0           ; If not within 10 pixels then
120         bhi.s   g_next           ;  try next invader
121         move.w  -(a0),d6         ;    Remove
122         move.b  -(a1),d7         ;      Dead
123         bsr     blank_out        ;        Invader
124         move.w  #-1,(a0)         ; Cancel invader from  future
125         move.b  #-1,(a1)         ;  appearances
126 ;        bsr     explode         ; More about this soon!
127         lea     zam,a3           ; Sound: See text - this is fun...
128         moveq   #17,d0           ;
129         trap    #1               ;
130 ;        bsr     inc_score        ; Future issue....
131         lea     inv_speed,a0     ; Check if this is the
132         cmp.b   #2,(a0)          ;   last invader
133         bne.s   notnew           ;
134         lea     invnext,a0       ; If yes, increase start pos.(lower)
135         addq.b  #8,(a0)          ;
136         bra     new_screen       ; Sets up new school of invaders
137 notnew  subq.b  #4,(a0)          ; If no, increase step speed
138         rts                      ;
139 g_next  dbf     d4,nextst        ; Tests next invader
140         cmp.b   #25,d7           ; Checks for mother ship hit
141         bhi.s   no_bus           ;
142         moveq   #nemalt,d7       ; If yes, then
143         lea     baddyx,a0        ;  these lines
144         move.w  (a0),d6          ;    delete the
145 ;        bsr     saver            ;       mother ship
146 ;        bsr     endbad           ;
147 ;        bsr     explode          ; Future issue....
148 ;        moveq   #39,d0           ; Increase score 39 times
149 ;morepts bsr     inc_score        ;
150 ;        dbf     d0,morepts       ;
151 no_bus  rts                      ;
152 no_hit  subq.w  #3,d6            ; Replots bullet after failing to
153         bra     plot             ;  explode by hitting anything
154 ;
155 ; Plots and unplots any bullet on the screen between on screen
156 ; movements of invaders, etc.
157 ;
158 rev_bul
159         move.w  bulposx,d6       ; These two lines establish x-cord of
160         bmi.s   no_rev           ; bullet & decide if it exists
161         move.b  bulposy,d7       ; Loads y-position
162         moveq   #9,d5            ; Set sprite number
163         bra     plot             ; Unplot or plot
164 no_rev  rts
165 ;
166 ; This routine tests the colour of a pixel specified by d6(x) and
167 ; d7(y) and returns the data in d2
168 ;
169 id_col
170         movem.l d6-d7/a0-a1,-(a7)
171         bsr     calc_addr        ; Branch to calc_addr
172         lea     mask_tab,a0      ; This routine utilises a table
173         add.b   d2,d2            ;  of masks which cover each of
174         add.w   d2,a0            ;   the bits in the two bytes used
175         move.w  (a1),d2          ;     in screen organisation
176         and.w   (a0),d2          ;       (see text)
177         movem.l (a7)+,d6-d7/a0-a1
178         rts
179 ;
180 ; These are sprite definitions and MUST be entered immediately below
181 ; the sprite definitions given in last month's issue
182 ;
183 ; Sprite # 8                 base
184         dc.w    $1818,$3c24,$7e5a,$e7bd,$db7e,$bdff,$7eff,$ffff
185 ; Sprite # 9                 bomb
186         dc.w    $0808,$1010,$0808,$1010,$0808,$1010,$0808,$1010
187  end
```

BOOKMARKS



As QL horizons broaden, so does its associated literature – Nicky Trevett explores this wider appeal.

The QL user manual might be comprehensive and packed with information and detail, but Stepen Morris thinks it might also bewilder the new QL owner who has never touched a computer before. So just for novice (and nervous users), he has written *Getting To Know Your Sinclair QL,* published by Duckworth.

This is a sort of 'pre-manual' manual which sets out to make the reader aware of the uses and potential of computers, de-mystifying the concepts and jargon along the way. It takes you carefully through the whole process of getting started, from unpacking the boxes and getting the microdrive out of its sleeve to taking care of the keyboard. It's real nuts and bolts stuff which tells you literally everything you ever wanted to know about coming face to face with a computer for the first time.

What is particularly likeable about the book is the way it ranges itself so emphatically on the side of the user. It is cautious, for example, about recommending the QL in the first place. The QL, it says, offers a sound investment for anyone thinking of buying a business computer who lacks previous experience of computers, but "whether it can become truly useful in the business world is yet to be seen".

The computer's good and bad points are ruthlessly dissected and all jargon is expurgated – not even 'hardware' and 'software' escape.

Perhaps best of all, the book makes no bones about the dangers of trusting to a computer ("relying on the computer can be disastrous") and details at inordinate length everything that could conceivably go wrong, especially the mistakes that can lead to the loss of a cartridge full of data and hours of painstaking work. All highly alarming, but it is meant to be, and too many books and manuals leave you to find this sort of thing out for yourself.

Towards the end, the spotlight is turned on software. A couple of chapters on programming and languages, and software applications (the Psion family), attempt to put you in the picture without duplicating the manuals. Again, a thoughtful approach for the novice.

This is a well-written and well thought-out little book for the beginner which mostly succeeds in filling the holes it perceives in the standard documentation, and is probably worth its £7.95 cover price.

Artificial Interface?

QL computing doesn't have to be all business and games. You can, for example, always try dipping a toe into the fascinating world of Artificial Intelligence as perceived by Tim Hartnell in his latest work on the subject, *Exploring Artificial Intelligence On Your Sinclair QL,* published by Interface and priced at £6.95. (if you find yourself thinking, funny, that title rings a bell, then you've probably come across the same book tailored for another computer. That is, a slight change to the front cover and modified program listings).

Artificial Intelligence (AI) is a vast, sprawling and at times controversial area of study which has been around for decades and only recently caught the imagination of the business world. Now home users have a chance to join in, and there's a great deal of entertainment to be gained – it can't really be put higher than that – from working through books like this and experimenting at a superficial level with some of the concepts.

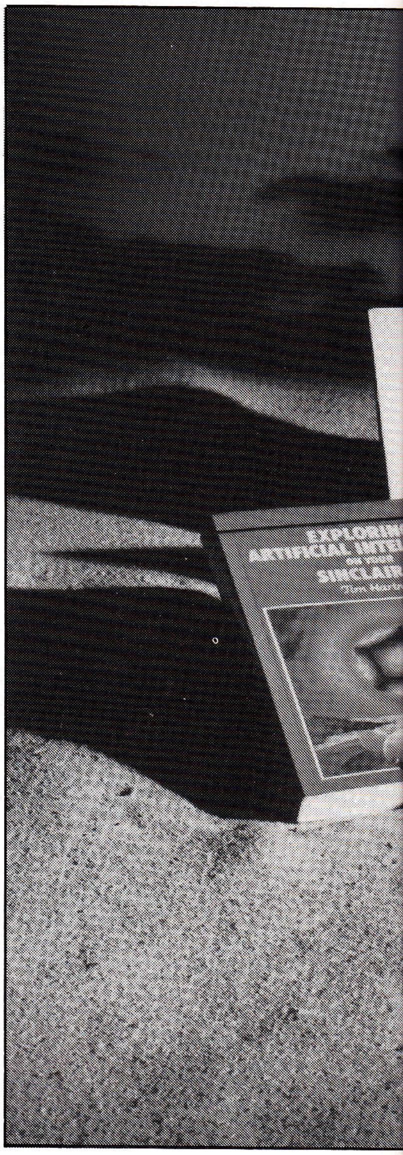
So what do you get for your £6.95? Like all of Mr Hartnell's books, this one is highly readable and quite painless, with the emphasis on games, which are used to demonstrate the AI techniques discussed. There's a lot of light-hearted discussion of the concepts of learning and reasoning – don't look for heavy philosophy or moral reflections – all heavily laced with the background and history of serious AI research, and conscientiously annotated with references to major works and theses.

Programs such as TicTac and Syllogy help show how computers can 'learn' and 'reason', to an extent, and there are versions of the famous trail-blazing Shrdlu, the program which manipulates building blocks (here called Blockworld). The inevitable Eliza-clone, called Doctor, is also included and claimed to be one of the most complete ever published in BASIC (or to be more precise the QL's SuperBasic).

The way in which a computer can be taught to search for and assess various related options, and decide on the best course of action, is demonstrated by a long and tortuous program called Snickers, a sort of cut-down checkers.

No book on Artificial Intelligence would be complete without a look at Expert Systems, a fast-developing offshoot from AI which is finding favour in the business world. Sure enough, we are referred to the well-known, highly successful Dendral program which is used worldwide to work out facts about molecular structures. Unfortunately, the program listings provided to show off expert system techniques are frivolous and disappointing.

This book is indeed all about having fun. You get a rapid, sweeping and easily digestible introduction to a highly complex but engrossing topic, with plenty of amusing programs to try out, but not much more. Nothing is explored to any great depth, and there is a lot left out – like the importance of programming languages such as LISP and PROLOG, which can be highly effective in the building of systems to 'learn' and make deductions. But anyone new to the subject is sure to be hooked.



All At C

With the release of 'C' compilers for the QL (GST et al), keen programmers casting about for information on the subject could try, in the absence of books devoted to C on the QL, two rather heavyweight guides from Prentice/Hall.

The C Programming Language by Brian Kernighan and Dennis Ritchie, priced at a staggering £22.95 (it's not even hard-backed), can at least boast the experience of the man who created C (Denise Ritchie) and implemented it on the Unix operating system on

BOOKMARKS

PHOTO BY TERRY BEDDIS

Digital Equipment Corporation's PDP-11 computers.

C is generally considered to be a general-purpose language which is not tied in to any particular application. At first glance it does not appear to have a great deal going for it, yet it is proving increasingly popular with professional programmers. Its lack of restrictions, and above all the fact that it is highly portable, have won it widespread favour.

Both *The C Programming Language,* and *Learning to Program in C,* by Thomas Plum, set out to show you how to program in C. They do it in a remarkably similar way. Both, for example, start with a tutorial-style introduction; Brian Kernighan and Dennis Ritchie assume some familiarity with basic programming concepts like variables, loops and functions, but Thomas Plum presupposes no programming experience at all.

Next, both books look at the features of C, covering them in much the same order and even numbering and subdividing the chapters in the same way. There are chapters on data types, operators and expressions, followed by statements and control flow, functions, pointers and arrays. *The C Programming Language* additionally looks into input and output and the Unix System interface, while *Learning to Program in C* helps to make your choice, *Learning to Program in C* is

Perhaps it simply goes to show how like-minded are the followers of C.

The two volumes share one other important underlying assumption – that the reader has access to Unix-based DEC PDP-II minicomputers (costing many thousands of pounds). These are not books intended for home users. Thomas Plum aims to turn the reader into a "competent programmer in a real software environment".

Having said that, there is still a great deal of information, presented in a practical way through actual programs, for anyone seriously interested in getting to grips with the C programming language. If it includes a chapter on software development.
less exhorbitantly priced – it costs £15.95.

Be Prepared

The book *Logic, Algebra and Databases* by Peter Gray is even more specialised. Published by Ellis Horward at £9.95, it "examines the applictions of logic programming and applicative programming to database through the database query languages". Quite so.

This is aimed at students, postgraduates and programmers who want to know more about database query languages, but anyone interested in logic and logic-based programs should find it of use. In particular, there is a great deal about the so-called fifth generation programming language PROLOG, and the book's extensive discussion of logic, inference and list processing fills in some of the holes left by introductory books like Tim Hartnell's Artificial Intelligence series.

The style might be academic, but for an academic book on logic the tone is pleasantly informal. To make much sense of the content you would need a certain amount of knowledge of a language like Pascal, which can manipulate records. It would also help if you had some grounding in logic; the book includes an introduction to the Predicate Calculus and the Lambda Calculus, which is best understood if you at least know what they are.

There is, however, some explanation of logic programming, followed by applicative programming and its development to functional programming, which is contrasted with methods used in languages like COBOL. Two classic database models are described (relational and Codasyl), and all the database query examples used are based on the same database – World Cup football matches – to point out the similarities and differences between languages.

If you're interested in logic, and in relating logic to database query, this could be for you.

Machine Code Tutorial

Adam Denning turns his attention to some 68000 practicalities and details modes of address.

One of the most important concepts we have to grasp is that of the addressing mode. This describes the way in which the processor accesses the data it manipulates, and it's a general rule that the more addressing modes a processor supports, the more versatile it is. The 68000 supports a wide variety, as you would expect for such an advanced design.

To meet the first few addressing modes, we'll write a little program to add two numbers (okay, it's not much fun, but it does lead the way).

If the two numbers we want to add are stored in memory locations $20000 and $20002, then we could write our program to add the two together like this:

```
    MOVE.W      $20000,D1
    ADD.W       $20002,D1
    MOVE.W      D1,$20004
```

What have we done here? The first instruction is a MOVE – *and* we know that it moves data from one place to another. The next part of the instruction (.W) signifies that we want to move a word of data (16 bits). Where does this data come from and where is it moved to?

The two operands after the mnemonic give the clues to those questions. The first operand, $20000, is known as the source, as this is where the 68000 will get the data from. The second operand, D1, must therefore be the destination. We can see that the destination is data register D1, but what exactly does the number '$20000' refer to in the source specification? Well, assuming that the use of $20000 in the MOVE instruction refers to the contents of that address. Whether we used the address as the source or the destination, using it in this form in a MOVE instruction always refers to the *contents* of the address, not the physical value of the address itself.

This is commonly referred to as absolute addressing, since $20000 is an 'absolute' address (meaning that it is not an offset from anything – $20000 really means *memory location* $20000).

The next instruction on our program is an ADD, with the prefix '.W' signifying the use of word-long operands once more. The source is again obtained using absolute long addressing, and the destination is a data register. Notice how the destination register is D1, just as it was in the previous instruction. You may think this will overwrite the value we've just moved there. Yes it will, but in a way which is totally predictable and under our control. The ADD instruction takes the value specified by the destination (which is the value we've just moved into D1 from location $20000), adds the value obtained from the source, and puts the result back into the destination.

This means two instructions have added our two numbers together and left the result in D1. We need some way of storing the result, so that we can look at it later, and use another MOVE instruction to put it into memory location $20004. Notice how this move uses exactly the opposite operations to our previous one – the source is found using data register direct addressing, and the destination is found using absolute long addressing.

So now we have the result of our addition in location $20004, but it isn't much use to us stuck up there. We need some way of looking at it, and the only way of doing this at the moment is to call the code from SuperBasic. In order to do this, we need to observe certain rules dictated by the SuperBasic interpreter. These tell us that data register D0 is used as an error return register, which means that it must be zero on return to Basic to return correctly.

There are lots of ways of setting a register to zero: we can use the special instruction CLR which does just that (CLR.L D0), or we can take the value of the register away from itself and put the result back into the register (SUB.L D0,D0), or we can use a useful little instruction called MOVEQ which only applies to data registers. This mnemonic stands for 'move quick', and it allows us to put any value between 0 and $FF into a data register in one quick instruction. We must be aware of certain facts, though. MOVEQ *'sign-extends'* the number we put into the data register, which means that any positive 8-bit number (ie, between $00 and $7F (0 and 127) is put into the entire 32 bits of the data register as just that value, while any negative 8-bit numbers (ie, between $80 and $FF (–128 and –1) cause all the bits from bit 8 to bit 31 to be set to 1, making $FF become $FFFFFFFF, $FE become $FFFFFFFE and so on. Using MOVEQ to put 0 into D0, then, actually sets all 32 bits of D0 to 0.

Our final addition program now looks like this:

```
    MOVE.W      $20000,D1
    ADD.W       $20002,D1
    MOVE.W      D1,$20004
    MOVEQ           #0,D0
    RTS
```

Notice how we put a '#' before the 0 in MOVEQ; this says that we are using the number zero rather than the contents of address zero. This is known as immediate addressing, as the number to be used immediately follows the instruction in memory.

To be able to use our program, we need to put the machine code numbers representing these instructions into memory, and also to POKE our numbers to be added into memory. This program will do the job:

```
100 RESTORE
110 addr = 131078
120 FOR a = 0 TO 20 STEP 2: READ n: POKE_W addr + a,n
130 REPeat loop
140 INPUT#0; 'First number:' !n1
150 INPUT#0; 'Second number:' !n2
160 POKE_W 131072,n1: POKE_W 131074,n2
170 CALL addr
180 PRINT#0; 'Result is:' !PEEK_W (131076)
190 END REPeat loop
200 DATA 12857,2,0 : REMark MOVE.W $20000,D1
210 DATA 53881,2,2 : REMark ADD.W $20002,D1
220 DATA 13249,2,4 : REMark MOVE.W D1, $20004
230 DATA 28672,20085 : REMark MOVEQ #0,D0 ; RTS
```

Try entering different numbers, even negative ones, for n1 and n2, and see if you can explain all the different results you get. You'll find that most things are sensible, and any that don't appear so at first can always be explained if we go back to two's complement numbers. Also, remember that we are adding words, so each number, including the result, is held in 16 bits.

Next month – more illustrative programs and a look at QDOS to see how it can be handled using machine code.

PRINTERLAND

ALL ORDERS DESPATCHED SAME DAY
SELLING PRINTERS SINCE 1980 !

DOT MATRIX

		EX VAT	INC VAT
BROTHER HR5		£133.00	£152.95
SHINWA CPA80		£185.00	£212.75
EPSON RX80		£185.00	£212.75
EPSON RX80F/T+	SPECIAL OFFER	£212.00	£243.80
EPSON RX100F/T		£347.00	£399.05
EPSON FX80		£314.00	£361.10
EPSON FX100		£425.00	£488.75

DOT MATRIX PLUS NEAR LETTER QUALITY

KAGA KP-810		£245.00	£281.75
CANON 1080A	SPECIAL OFFER	£245.00	£281.75
KAGA KP-910		£340.00	£391.00
CANON 1156A	SPECIAL OFFER	£340.00	£391.00

DAISY WHEEL

QUENDATA 1120	£225.00	£258.75
JUKI 6100	£325.00	£373.75
EPSON DX100	£356.00	£409.40

COLOUR PRINTERS

EPSON JX-80	SPECIAL OFFER	£450.00	£517.50
CANON PJ1080A		£425.00	£488.75

QL RS232 INTERFACES

MIRACLE SYSTEMS	£32.50	£37.37
SINCLAIR QL	£334.00	£384.10

FREE with all Printers. QL users Printers Guide. Getting the best from your printer is not easy !

We supply a FREE Booklet with all printers which explains how to obtain all the features available on your printer. Without this you could waste many hours. Buy else where and you'll pay far more and get far less.

DISK DRIVES

We stock and use ALL drives and interfaces available including MEDIC, CUMPUTAMATE and MICRO PERIPHERALS. For helpful impartial advice and best prices contact us.

MICRO P DISK INTERFACE	£99.00	£113.85
Q DISK INTERFACE	£129.57	£149.00
3½" SINGLE DISK DRIVE 75 MB	£159.00	£182.85
3½" DUAL DISK DRIVE 5 MB	£298.00	£342.70
5½" SINGLE DISK DRIVE .75 MB	£169.00	£194.35
5½" DUAL DISK DRIVE 1.5 MB	£347.83	£400.00

MONITORS

PHILIPS 7502 GREEN	£75.00	£86.25
MICROVITEC CUB 653	£220.00	£253.00

FULL PACKAGE OFFER

Sinclair QL. Microvitec Cub 653
Epson RX80 F/T.
Price includes all leads and interfaces.
(Other printers also available)

REC RETAIL (INC VAT)	£1070.00
OUR PRICE (INC VAT)	£899.00

SHOWROOM OPEN 9.00am to 6.00pm DAILY
9.00am to 1.00pm SATURDAY
Delivery Printers/Monitors (Securicor) £10.00 Other £1.00

Printerland, Unit 27, Estate Buildings, Railway Street
Huddersfield HD1 1JP

Telephone Huddersfield 514105/512037

CLASSIFIED

QL COMPUTER DESK

Throw away those silly little feet from Sinclair! Support your QL at recommended slope. Allows full access to MDVs and sockets. Soft rest for wrist. Matching black finish. £11.00 all inclusive by post.

BROOMSPRING
99 Broomspring Lane, Sheffield. Tel: 0742 737000

HI-RES SCREEN DUMP FOR YOUR QL

DEVELOPED BY LOTUS-SOFT

★ **New enhanced version 2.0 with windowing facility** ★

Features: ☆ Epson or compatible printers. ☆ 100% 68000 machine code. ☆ Free Hexpoke program. ☆ Links in with QDOS and SuperBasic (new Keyword added).

★ USER DEFINABLE PRINTING WINDOW & PRINTER POSITION ★

☆ Loaded as resident procedure. ☆ Full error messages. ☆ ESCAPE key function.

ONLY £9.95 INC.

Send Cheque/PO/Access/Visa for £9.95 to:

Positron Computing, 15 Central Precinct, Llanelli,
Dyfed, SA15 1YF. Tel: 0554 759624 (SAE for details)

INVESTORS!

Track your shares with "STOCKMARKET MANAGER" from *Portfolio Software*

Facilities include: • Simple entry of purchases, sales, prices etc. • Portfolio valuation – analysed by investment type-comparison of performance against market. • Calculation of % return on each investment. • Assessment of Capital Gains Tax liability. • Handles all investment types, including Traded Options.

Orders/Cheques to: **PORTFOLIO SOFTWARE, PO Box No. 15, London SW11 5RP**

No. of copies

Please send: "STOCKMARKET MANAGER" @ £39.95 incl. p&p. □
Information sheet (please enclose SAE) □

Name ....
Address ....
.... QLU/May

THREE 100% MACHINE CODE, FULL GRAPHIC, VERY FAST MOVING, ARCADE STYLE GAMES

1. Merry Muncher
2. Fire Tower
3. Advance Invaders

BLAIN SOFTWARE
8 Berkeley Close, Staines, Middlesex, TW19 6ED

DATAMANAGEMENT

— leading the way with QL software

SBUTIL	(£9.00)	—Possibly the last word in Basic programming utilities.
BACKUP	(£8.00)	—An independent, fast and reliable backup to/from Mdv or Disk.
TERMINAL	(£12.50)	—A full featured terminal package.
CHARGEN	(£10.00)	—A comprehensive character generator.
SBEXTRAS	(£6.00)	—Many extra features and procedures to add to SuperBasic.
QLDIARY	(£10.50)	—A SuperBasic Diary package.
QLLIFE	(£6.00)	—This M/L version for the QL's graphics is the prettiest to date.
DATABOSS	(£10.50)	—A user definable Database in SuperBasic.
MAILBOSS	(£10.50)	—A special version of DATABOSS for handlig those mailing lists.
QLFED	(£12.50)	—If you ever needed to modify a file this is your program.
ANIMAL	(£10.50)	—A game of 'Think of an Animal' for Archive — very instructive.
PROJECT	(£8.00)	—A fast, pretty 3D presentation of various equations.
FM	(£12.50)	—A file management package for Disk/Mdv. Fast and effective file control.

All 100% m/language, multi-tasking unless SuperBasic is stated. Prices shown include VAT & P/P, but add £1.99 per order for a mdv cartridge.

Cheques or P/Os to **DataManagement:** 12 Larch Way, Haxby, YORK, YO3 8RU or telephone (0904) 760351 (anytime) for our latest list, which is growing daily.

Have **you** written any good **QL** software. We are looking for the **best** software and are prepared to pay the **best** royalties.

Our latest list and newsletter explains in much more detail. Write or phone now for your copy. We also have a number of 'free bees' such as **SCREENCOPY, NEATLIST, DIRMAP, & FREEMEM** which are distributed with orders. However, send a cartridge and a large SAE and we will send copies with your list and newsletter.

Drives Apart–High Capacity Low Cost Disk Drives for QL and BBC Micro`s

With so many disk drives on offer, choosing can be a somewhat confusing and difficult exercise. On the surface they all seem to perform and look the same. Of course that's not the case and what it all boils down to at the end of the day is the data storage reliability and the number of bytes of disk storage you get for every £ spent.

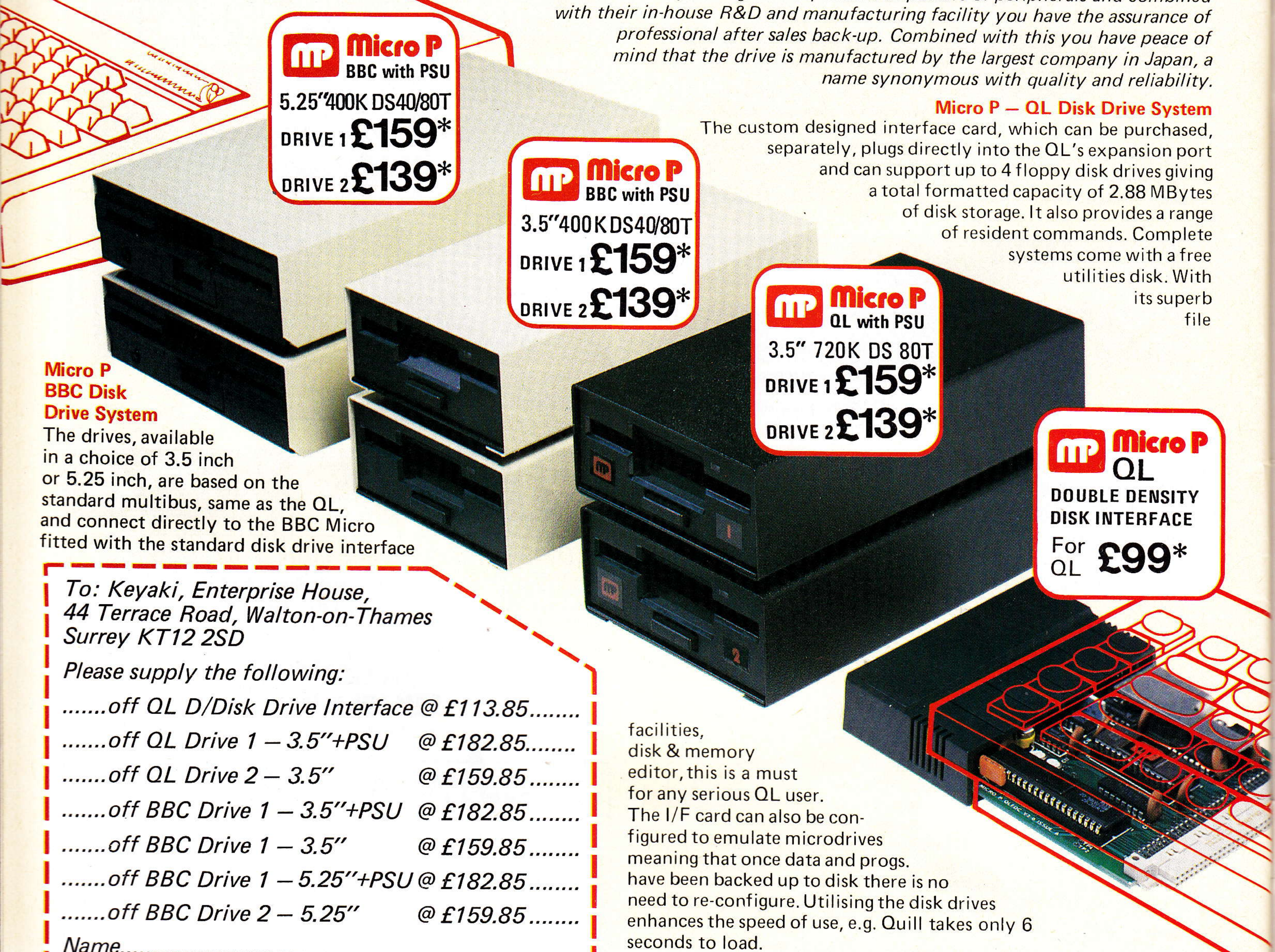
Take the new range of professional 3.5 inch and 5.25 inch disk drive systems from Micro P. They are designed for use with QL and BBC Micros, housed in a rugged metal case, essential for screening from stray electrical fields which can reduce data storage reliability, and have their own power supply so as not to drain precious power from your micro needed for other less sufficient peripherals.

Most important of all you get in excess of 452 bytes of disk storage per £ for the QL and 251 bytes for the BBC.

A further added bonus of the Micro P drives is that they are available in the new 3.5 inch format which is undoubtedly set to become the standard.

Not only do 3.5 inch drives take minimum space, quite often at a premium for the micro user, but they also use easily transportable pocket sized media and are automatically protected from dust and finger prints.

Micro P are one of Europe's largest independant importers of peripherals and combined with their in-house R&D and manufacturing facility you have the assurance of professional after sales back-up. Combined with this you have peace of mind that the drive is manufactured by the largest company in Japan, a name synonymous with quality and reliability.



Micro P
BBC with PSU
5.25″400K DS40/80T
DRIVE 1 £159*
DRIVE 2 £139*

Micro P
BBC with PSU
3.5″400K DS40/80T
DRIVE 1 £159*
DRIVE 2 £139*

Micro P
QL with PSU
3.5″ 720K DS 80T
DRIVE 1 £159*
DRIVE 2 £139*

Micro P
QL
DOUBLE DENSITY DISK INTERFACE
For QL £99*

Micro P — QL Disk Drive System

The custom designed interface card, which can be purchased, separately, plugs directly into the QL's expansion port and can support up to 4 floppy disk drives giving a total formatted capacity of 2.88 MBytes of disk storage. It also provides a range of resident commands. Complete systems come with a free utilities disk. With its superb file facilities, disk & memory editor, this is a must for any serious QL user. The I/F card can also be configured to emulate microdrives meaning that once data and progs. have been backed up to disk there is no need to re-configure. Utilising the disk drives enhances the speed of use, e.g. Quill takes only 6 seconds to load.

Micro P BBC Disk Drive System

The drives, available in a choice of 3.5 inch or 5.25 inch, are based on the standard multibus, same as the QL, and connect directly to the BBC Micro fitted with the standard disk drive interface

*To: Keyaki, Enterprise House,
44 Terrace Road, Walton-on-Thames
Surrey KT12 2SD*

Please supply the following:

.......off QL D/Disk Drive Interface @ £113.85........
.......off QL Drive 1 – 3.5″+PSU @ £182.85........
.......off QL Drive 2 – 3.5″ @ £159.85........
.......off BBC Drive 1 – 3.5″+PSU @ £182.85........
.......off BBC Drive 1 – 3.5″ @ £159.85........
.......off BBC Drive 1 – 5.25″+PSU @ £182.85........
.......off BBC Drive 2 – 5.25″ @ £159.85........

Name..
Address..
...Postcode................
Telephone..
Tick box for details of Instant Credit up to £1000
I enclose my cheque no. ....for £..........
Please charge my Access/Barclaycard No.
□□□□□□□□□□□□□□□□
Signature: ....

VISA

*Prices do not include VAT

☎ **Telephone Credit Card Orders welcome.**
Personal Collection welcome.
Open Monday to Saturday – 9.00 a.m. until 5.30 p.m.

THE KEYAKI 100% GUARANTEE
– Full 1 yr. no quibble warranty
– 14 day Money Back Guarantee
–Delivery within 7 days subject to availability.

£1000 instant credit, writtent details on request.

KEYAKI

Enterprise House 44 Terrace Road, Walton-on-Thames,
Surrey. KT12 2SD Tel: 09322 42777

TRADE & EDUCATIONAL ENQUIRIES WELCOME

Micro P is a registered trade name
Keyaki is a trading name of Micro Peripherals Ltd

QL is a registered trade name of Sinclair Research Ltd
QUILL is a registered trade name of Psion Ltd

▼ MICRODRIVE EXCHANGE ▲

Each month, for a trial period, this column will contain details of readers' programs that we are able to offer on microdrive.

In return for a small administration charge (per program – including a royalty for the author), we will copy onto blank microdrives any or all of the featured programs.

Each program will be a direct copy of the published listing, or an extended version of that listing where the program in question was too long to print in full (programs for which an abridged version has been published are marked with an asterisk).

It must be stressed that we are not selling the software itself, nor providing any guarantee that it performs any particular function (though we do check every program that is to appear in *QL User*), we are merely offering a service to readers who wish to obtain *QL User* programs on drive rather than by typing them in straight from the page.

HOW TO ORDER

Listed below are programs which have appeared as listings inside *QL User*. Each entry includes a brief description, the author's name, approximate size in mdv sectors and the price.

Prices range from £1 to £5 dependent upon the size of the program and the time it takes to copy onto microdrive.

To the right of each program entry is a small box, which you should mark with a bold cross if you want to order that program.

Once you have put a cross next to all the programs you wish to have copied onto microdrive, simply complete the rest of the order form and send it along with you PO/cheque AND BLANK FORMATTED DRIVE to:

QL User, MICRODRIVE EXCHANGE, Priory Court,
30-32 Farringdon Lane, EC1R 3AU.

If you wish us to supply the drive, please add an extra £2.50 for every drive required and mark the order form appropriately.

ORDER FORM

Author	*Language*	*Program Name*	*Price*	*Issue*	*Size*
Richard Cross	(MB)	Function Key Definer	£2.00	May	20 □
Programmable function keys just like on the BBC					
Rob Miles	(B)	3Dscapes	£1.00	May	20 □
Isometric Perspective representations of 3D Surfaces					
Shergold & Tose	(B)	* Golf	£2.00	May	30 □
From fairway to green on 50 different courses of varying difficulty					
Williams & Holliday	(AO)	Paladin	£5.00	Apr	80 □
The basis of our games programming series – a space invaders type game written entirely machine code					
Richard Cross	(MB)	Sprite Animation	£2.00	Apr	32 □
A subtle blend of machine code and SuperBasic that produces a versatile sprite designer and high speed animator					
Steve Deary	(B)	Pacman	£1.00	Mar	20 □
A reasonably fast rendition of the famous arcade favourite					
Adam Denning	(AO)	File Probe	£1.00	Mar	2 □
Machine code routine utility to dump the contents of a file to any device in hex. Reveals any hidden control characters for all to see					
QL User	(B)	Tape Utility	£3.00	Feb	15 □
File utilities which permit selective back-ups and deletions – in excess of a hundred files may be manipulated					
Andy Carmicheal	(B)	Hi-tech Sort	£1.00	Feb	10 □
Highly efficient recursive sort based on C Hoare's famous Quicksort method					
PJ Smith	(B)	* DIY Adventure	£1.00	Feb	70
A skeleton framework where you simply have to slot in the details to create your bespoke adventure					
Mike Newport	(B)	Psuedo Editor	£1.00	Feb	20 □
An ingenious half-way house between a full screen editor and the QL's somewhat primitive line editor					
Adam Denning	(AO)	Multitasking	£2.00	Dec/Jan	15 □
Two digital clock programs and one alarm. All will happily multitask on the QL					

B = SuperBasic, AO = Assembler + Object Code (ready to run), MB = Machine Code + Basic loader

Name____

Address______________________________________

____

No of programs ordered .... Total cost £...........

Total sects
(max 200 per drive) ....

No of drives sent ....

No of drives required .... @ £2.50 each £...........

Plus postage & packing £ 0.75

TOTAL TO BE SENT £...........

Please copy onto microdrive the programs above which I have indicated with a cross. I enclose a cheque/PO to the value of £............ (made payable to *QL User*).

I understand that *QL User* only undertakes to SUPPLY these listings (copied onto microdrive) and accepts no liability for their operation as defined by the author. Neither can *QL User* supply additional information about any of the listings other than that originally printed in the magazine (any article reprints required must be ordered separately at £1 each inclusive of post and packing).

INSTA

Each month this directory is updated with new products and information.

If you or your company are currently manufacturing hardware or supplying QL software and would like to be included within this directory, just send details to 'QL User Reference Chart', Dept SE, QL User, Priory Court, Farringdon Lane, EC1R 3AU.

HARDWARE INDEX

Monitors

Citadel Products Ltd
01 951 1848

MBS Data Efficiency
0442 60155
Kaga

Microvitec PLC
0274 390011

Microworld Computer & Video Centre
0293 545630/0273 671863
Microvitec, Philips, Sinclair Vision

Opus Supplies Ltd
01 701 8668
JVC

Printerland
0484 514105/687875

Strong Computer Systems
0267 231246
Microvitec, Philips

Technomatic Ltd
01 208 1177
Microvitec, Kaga

Viglen Computer Supplies
01 843 9903

Zeal Marketing Ltd
0246 208555
Microvitec, Philips

Printers

Datasystems
01 482 1711
Star

MicroPeripherals
0256 473232
Canon

Microworld
0293 545630/0273 671863
Kaga, Epson, Smith Corona, Shinwa, Microperipheral, Quen-data, Juki

Printerland
0484 514105/687875
Epson, Brother, Kaga, Canon, Juki

Strong Computer Systems
0267 231246
Brother, Shinwa, Epson, Kaga, Mannesman Tally, Canon, Daisystep, Smith Corona

Technomatic Ltd
01 208 1177
Epson, Kaga, Juki, Brother

Twickenham Computer Centre
01 891 4991
Kaga, Ensign, Canon

Viglen Computer Supplies
01 843 9903

Zeal Marketing Ltd
0246 208555
Brother, Epson, Canon, Daisystep

Interfaces

Cambridge Systems Technology
0223 323302

Care Electronics
0923 777155

Computamate Data Products
0782 811711

Miracle Systems Ltd
0272 603871

Printerland
0484 514105/687875

Sigma Research
231 Coldhams Lane,
Cambridge

Technology Research Ltd
0784 63547

Transform Ltd
089 283 4783

Zeal Marketing Ltd
0246 208555

Disk Systems

Computamate Data Products
0782 811711

Compware
0270 582301

CST
0223 323302

Medic Data Systems Ltd
0256 460748

MicroPeripherals
0256 473232

Printerland
0484 514105/687875

Quest
04215 66488

Strong Computer Systems
0267 231246

Zeal Marketing Ltd
0246 208555

Modems

Compak Data
13 Beechwood Road, Uplands,
Swansea

NT ACCESS

Medic Data Systems Ltd
0256 460748

Strong Computer Systems
0267 231246

Memory Expansion

Eprom Services
0532 667183

Medic Data Systems Ltd
0256 52703

PCML Ltd
0372 67282/68631
QL + RAM cards

Quest
04215 66488

Simplex Data Ltd
01 575 7531

Extras

Action Computer Supplies
01 903 3921
Mains spike eliminator

Eidersoft
01 478 1291
Quicksoft 11 Joystick

Management Science Ltd
17 West Hill, London
QL Case

Power International
0705 756715
Mains spike eliminator

Sigma Research
231 Coldhams Lane,
Cambridge
Joystick Adaptor

Sinclair Research
0276 686100
Power Supplies

Transform Ltd (Dept QL)
089 283 4783
QL Dust Cover, Microdrive Storage Box, RS232 lead

Viglen Computer Supplies
01 843 9903
Printer stand, disks, disk drive

Zeal Marketing Ltd
0246 208555
Printer peripherals

SOFTWARE INDEX

Utility Programs

Accountancy Software of Torquay
Sinclair Research
QL Cash Trader

Computer One
0223 86216
Pascal, Forth, Assembler, Typing Tutor, Monitor

Digital Precision
91 Manor Road, London E17
5RY (mail order)
QL Sprite Generator, QL Super Monitor

GST Computer Systems
0954 81991
QL Assembler, 68K-OS, c compiler

Harcourt
Sinclair Research
QL Touch 'n' go

Metacomco
0272 428781
Assembler, BCPL, Lisp

MicroAPL
01 622 0395

Positron Computing
0554 759624
Hi-res Screen Dump

Printerland
0484 514105/687875
Metacomco Assembler

Psion
01 723 9408/0553
Quill, Abacus, Easel, Archive

Quest
04215 66488
Business Accounts

Sinclair Research
0276 686100
QL Entrepreneur, QL Project Planner, QL Decision Maker

Strong Computer Systems
0267 231246

TDI Software Ltd
0272 742796
USCD Pascal, USCD Fortran, Advanced Development Toolkit

TR Computer Systems
093 924 621
QL Payroll

Games

Digital Precision
91 Manor Road, London E17
5RY (mail order)
QL Super Backgammon

Eidersoft
01 478 1291
QL Art

Games Workshop
01 965 3713
D-Day

Printerland
0484 513105/687875
Psion Chess

Psion
01 723 9408
Psion Chess

Sinclair Research
0276 686100
Psion Chess

Talent Computer Systems
041 552 2128
The Lost Kingdom of ZKUL, WEST, GraphiQL

BOOKS

Publishers

Adder
0223 277050

Century
01 434 4241

Collins
01 940 7070

Duckworth
01 485 3484

Granada
01 493 7070

Hutchinson
01 387 2811

Interface
9-11 Kensington High Street,
London

McGraw Hill
0628 23431

MicroPress
0892 39606

Melbourne House
01 940 6064

Prentice/Hall
0442 58531

Sunshine
01 437 4343

Book Titles

68000 Assembly Language Programming
Kane, Hawkins, Levanthal
(Osborne/McGraw Hill £19.50)

QL Gamesmaster
Ewbank, James & Gee
(Collins £7.95)

Understanding the 68000
Heller
(Century Communications £7.95)

Get More from the Epson Printer
Curran
(Collins £7.95)

A QL Compendium
Gandoff & Kinge
(Addison-Wesley £7.95)

QL Handbook
Hartnell
(Interface £7.95)

Professional & Business Uses of the QL
Lewis
(Collins £7.95)

BASIC Programming on the QL
Cryer & Cryer
(Prentice/Hall £7.95)

Quick QL Machine Language
Giles
(Melbourne House £7.95)

QL Abacus
Spottiswoode
(Century £8.95)

The Working Sinclair QL
Lawrence
(Sunshine £6.95)

Quantum Theory
San, Katan & Rockman
(Century £5.95)

QL SuperBasic
Wilson
(MicroPress £6.95)

QL Assembly Language Programming
Opie
(McGraw Hill £12.95)

QL Games Compendium
Hartnell
(Interface £5.95)

The QL Book of Games
Hurley & Virgo
(MicroPress £6.95)

Introduction to SuperBasic on the Sinclair QL
Meadows
(Hutchinson £6.95)

Advanced Programming with the Sinclair QL
Gandoff
(Hutchinson £6.95)

QL Quill
Simon & Spottiswoode
(Century £7.95)

QL Easel
Spottiswoode
(Century £7.95)

QL SuperBasic
Berk
(Granada £6.95)

Assembly Language Programming on the Sinclair QL
Pennell
(Sunshine £7.95)

Developing Applications on the Sinclair QL
Grace
(Sunshine £6.95)

Artificial Intelligence on the Sinclair QL
Brain & Brain
(Sunshine £6.95)

Desk-Top Computing with the Sinclair QL
Miles
(Hutchinson £6.95)

Introducing the Sinclair QL
Marshall
(Hutchinson £6.95)

Word Processing with the Sinclair QL
O'Reilly
(Hutchinson £6.95)

CLASSIFIED

QL FOR BUSINESS

OWL Computer Training

A programme of linked courses.
Introductory & Advanced Applications.

044 282 7302
FREEPOST, TRING, HERTS. HP23 4BR.

High speed resident m/c functions and procedures

SORT integers/Floating/Strings. Like lightning sort 1000 random FP no's ‹1.3 secs.

FILE LD/SV/RH/SH/SZ/MN/ASC/DEC/HEX/HEAP/HEAP–DA ›1500 lines of fully documented assembly code.

+ MORE + MORE + MORE + MORE + MORE + MORE

FILEDUMP File handling utility. LOAD/FIND/CHANGE/ MODIFY/SAVE/PRINT/ABS/INFORM etc ›80 files in memory at once. ›500 lines of code.

Both progs professionally produced and documented. For object code + fully documented source listing send your cartridge + GBP 9.95 or USD 11.50 to:
T:mi SOFTRONIC, Suvikuja 3B14, 02120 Espoo 12, Finland.

JOYSTICKS FOR QL

Plugs straight into control port 1 or 2.
NEEDS NO INTERFACE OR ADAPTER – £7.99 each or £15 for two.
Add 50p p&p.
Two joysticks plus CONCENTRATION (an addictive 1-4 player game) on microdrive cartridge for only £23.95 inclusive. *Game only: £10.95 inclusive.*

**Send to: COMPUTER SUPPLIES
146 Church Road, Boston, Lincs. PE21 0JX.**

WDSoftware

FOR THE QL:

WD Utilities (flp1) **(£10 on 5.25″ Disc)**
Improved version for the CST/Computamate Floppy Disc System. Tested with dual discs and four extra microdrives, more than 100 files on a disc without overflowing screen and backed-up with one command. The FAST way to move your (unprotected) software on and off disc! State 40 or 80 track when ordering.

WD Utilities (3rd ed) **(£7.50 on Microdrive)**
Up to 60 files for one-screen DIRectory, one-key LOAD, PRINT or COPY (namesakes protected). Prune old files to release space (one key DELETEs a file). Easy multiple FORMATting for extra reliability. Toolkit to make dated, numbered modules in program development. Control 6 extra microdrives (add on Spectrum ones)!

Ref QL (1st ed) **(£1 as an extra, £3 alone [disc or mdv])**
100+ useful QL references in ARCHIVE file.
Prices include postage in Europe.
WDSoftware, Hilltop, St Mary, Jersey. Tel: (0534) 81392

QL FORTH

Full implementation of 79-STANDARD FORTH with double number extensions. FORTH is much faster than BASIC and uses less memory. System includes a full screen editor, QDOS calls, QL graphics, utilities and many extensions to the standard. All features documented.

£15 from:

R. E. Jackson, 15 Edgehill, Llanfrechfa, Cwmbran, Gwent NP44 8UA

Prove to your friends that your computer is really useful.
Plan the garden or the house with our data banks and programs on shrubs and house plants. ONLY £11.95

Shrub bank and planner £11.95
House plant bank and planner £11.95

Name ....
Address ....
....
.... **Postcode** ....

For details or payment by credit card 097 423223
To: SuperPlant Software, Llangeitho, Tregaron, West Wales, SY25 6QY

QDISC PRICES FALL

	Single	Dual
3.5″ or 5.25″ 360K System	£308	£475
3.5″ or 5.25″ 720K System	£349	£529

QDISC I/F alone £149. Ten 5.25″ disks in plastic case £29.95
QDISC "Systems" include, power supply, all cables, typeset manual, VAT & delivery.
Telephone for prices of MEDIC DATASYSTEMS QL Disc Drives

Microvitec CUB with cable	£265	Microdrive '4 packs' in wallet	£9.99
Zkul or West Adventures	£19	ADDER Macro Assembler	£34.95
Psion Chess	£19	Metacomco Assembler	£39.95
QL Art Graphics Design	£15.95	Computer One Assembler	£29.95
Sinclair Toolkit	£24.95	Computer One Monitor	£24.95
Sinclair Cash Trader	£69.95	HISOFT MONQL	£24.95
QL Serial Cable	£15.95	Compware Toolkit	£14.95
Computer One FORTH	£39.95	Computer One Pascal	£39.95

All prices include VAT and delivery

COMPWARE

**57 REPTON DRIVE
HASLINGTON, CREWE CW1 1SA
TEL: (0270) 582301**

SPACE TREK

A **real-time text adventure** set in the distant future. Navigate and explore the galaxy, seeking out and boarding alien spacecraft while trading with friendly planets for weaponry and supplies.
£7.99 inc. p&p and full documentation.

New **GRAPHICS VERSION** now available, still only **£7.99 or £9.99 for both.**

Cheque or P.O. to:
Swansoft, 164 Vicarage Road, Morriston, Swansea SA6 6DT.
(Orders dealt with promptly)

☆ ☆ FANTASIA ADVENTURE ☆ ☆

An original and demanding text adventure for the QL set in an unpredictable and dangerous world of ancient cities, complex cave systems and harsh deserts. Features include:
● 200+ locations, ● detailed descriptions, ● save/restore game facility, ● real-time problems, ● a 'help' feature, ● comprehensive instructions, ● hint sheet on receipt of a s.a.e. ALL THIS FOR ONLY £8.50 (p&p free).

Send cheque/postal order to:
S.B. SOFTWARE, 20 ST. NICHOLAS ST., DISS, NORFOLK

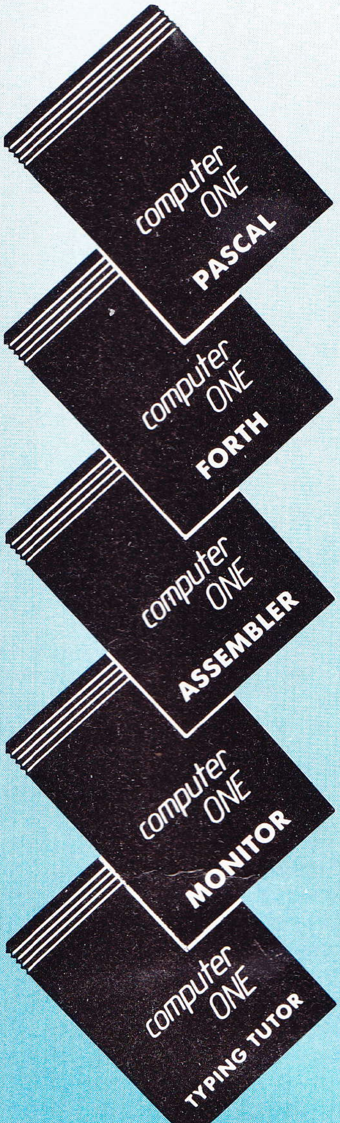
Terminal Emulator with file transfer & VT52 emulation	£15.00
68000 Assembler/Editor	£12.95
3 Games on 1 cartridge—sprite Shooting (M/Code), Lander (M/Code), Moon landing	£4.95

Cheques, P.O.s to: **QCODE,** 42 Swinburne Road, Abingdon, Oxon OX14 2HD.
or tel: 0235 28359 any time

AD INDEX

QL POWER

AT YOUR FINGERTIPS FROM COMPUTER ONE



PASCAL £39.95

Computer One PASCAL is a powerful implementation of this classic programming language. Produced specifically for the QL, this package provides a professional and highly educational programming system. Complete with comprehensive 100 page manual, this is the ultimate language package that no QL enthusiast can do without.

FEATURES: ★ Full screen editor ★ QL graphics and sound extensions ★ Example programs ★ Complete 100 page User Guide ★ Produce stand-alone programs ★ Full implementation close to ISO-standard ★ Typically 15× faster than BASIC

FORTH £39.95

A full implementation of the latest FORTH-83 standard with full integration to QDOS. This compiled language runs very fast and allows you to produce fast machine code application programs for your QL.

FEATURES: ★ FORTH-83 system ★ Forth screen editor ★ 100-page manual ★ Produce stand-alone programs ★ QL-graphics and sound extensions ★ Example FORTH programs ★ FORTH macro-assembler ★ Multitasking extensions with many powerful features

ASSEMBLER £29.95 NEW: Now includes FREE linker

COMPUTER ONE Assembler is a comprehensive assembler program toolkit comprising a full 68008 assembler and full screen editor. This is a professional multitasking assembler and is written in machine code for speed and compactness.

FEATURES: ★ 68008 Assembler ★ Full syntax checking ★ Full screen editor ★ Assembler linker ★ Comprehensive User Manual ★ Integrates to Pascal and Superbasic ★ INCLUDE file facility ★ Full assembler listing

MONITOR £24.95 (£19.95 when purchased with the Assembler)

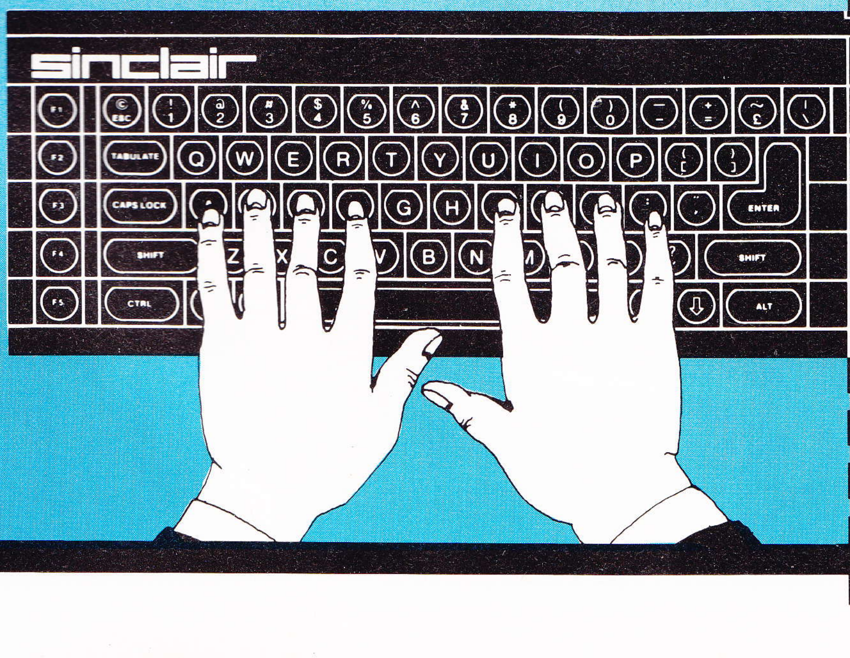
The Computer One MONITOR is a powerful multitasking, multi-window debugging monitor. The MONITOR is an invaluable software tool for anyone developing software for the QL and is ideal for use in conjunction with the Computer One Assembler.

FEATURES: ★ Memory disassembly and display ★ Trace code in RAM or ROM ★ Breakpoints and watchpoints ★ Full channel management ★ User defined screen layout ★ Set and display registers ★ Alter memory ★ Move and search memory ★ Full job control ★ Ability to monitor several jobs simultaneously

TYPING TUTOR £19.95

Let your Sinclair QL teach you to type! Computer One's Typing Tutor is a fast, easy and enjoyable way of learning true touch-typing on your QL.

FEATURES: ★ Self-paced tutorials ★ Speed and accuracy diagnosis ★ Interactive colour graphics ★ User Guide



computer ONE

SCIENCE PARK, MILTON ROAD, CAMBRIDGE

Name ______________________

Address ______________________

____

Signed ______________________

	Tick Order
PASCAL	□
FORTH	□
ASSEMBLER	□
MONITOR	□
TYPING TUTOR	□
Please send further details	□

Access

Method of Payment-
by cheque or P.O. (no cash) and made payable to COMPUTER ONE, or by charge card.

Access Card No.
□□□□□□□□□□□□□□□□

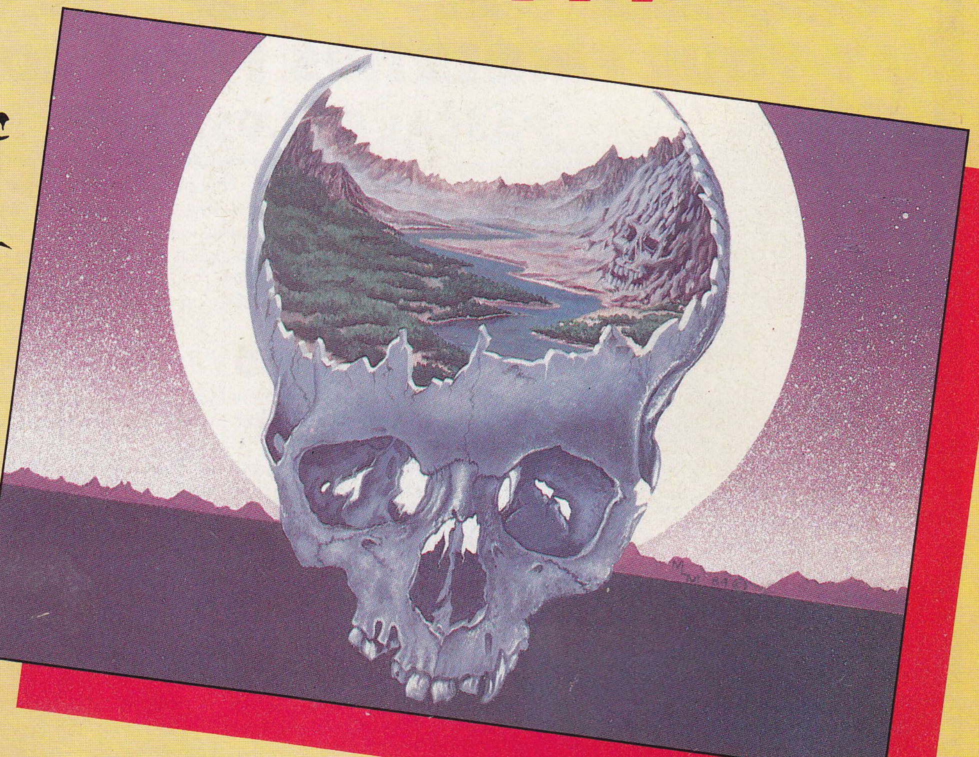
Total amount £ ______________________
enclosed with order

QL/5/85

Credit card sales: UK buyers only.
Please allow 28 days for delivery.

Inc. VAT & p&p.

The first QL adventures from TALENT!



The Lost Kingdom of ZKUL

Explore the ancient dwelling-place of the Dwarves – where the Wizard guards the last precious secret.

A classic and complex text adventure with hundreds of locations and a huge vocabulary. Set in real-time with traps, tasks, puzzles and mazes – and a special note-pad feature to aid you.

"the quality of the game is superb"
Micro Adventurer

PRICE
£19.95
+ 50p postage and packing

WEST

You are on the track of a notorious gang of bank robbers who have gone to ground near an abandoned mining town deep in Indian territory. Your task – to outwit and outgun the robbers, collect as much loot as possible and piece together clues on how to escape.

A demanding and exciting text adventure for the experienced games-player, using over 200 words and as many phrases. There are over 130 locations to search – not all of them easy to find. Events happen in real-time outside your control – Indians charge, rattle-snakes slither past and robbers appear and shoot at you. How long can you survive?

PRICE
£19.95
+ 50p postage and packing
(Also on CBM 64, BBC & Electron)

Available from selected branches of:
John Menzies
WHSMITH

TALENT
COMPUTER SYSTEMS

CURRAN BUILDING, 101 ST JAMES ROAD, GLASGOW G4 0NS
24 HOUR CREDIT CARD HOT-LINE 041-552 2128